A FRAMEWORK FOR DESIGNING DATA-DRIVEN OPTIMIZATION SYSTEMS FOR NEURAL MODULATION: WITH APPLICATIONS IN OPTOGENETIC AND ELECTRICAL BRAIN STIMULATION

A Dissertation
Presented to
The Academic Faculty

by

Mark J. Connolly

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
Biomedical Engineering

Georgia Institute of Technology
Emory University
May 2021

A FRAMEWORK FOR DESIGNING DATA-DRIVEN OPTIMIZATION SYSTEMS FOR NEURAL MODULATION: WITH APPLICATIONS IN OPTOGENETIC AND ELECTRICAL BRAIN STIMULATION

Approved by:

Dr. Robert E. Gross, Advisor
Department of Neurosurgery
Emory University School of Medicine

Dr. Annaelle Devergnas
Department of Neurology
Emory University School of Medicine

Dr. Babak Mahmoudi
Department of Bioinformatics
Emory University School of Medicine

Dr. Svjetlana Miocinovic
Department of Neurology
Emory University School of Medicine

Dr. Christopher Rozell
Department of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: April 23, 2021

To the animals that made the ultimate sacrifice to make this research possible.

ACKNOWLEDGEMENTS

This work would not have been possible without an absolutely amazing team of advisors, mentors, colleagues and students. Most critically, my advisor, Dr. Robert Gross and my thesis committee have provided insightful guidance as I made my way through each of the major milestones of this degree. On a day-to-day basis I was able to rely on the larger Gross Lab, to help put out the figurative (and literal) fires I may have started in the lab. Additionally, the success of this work depended on the tireless efforts of postdocs, grad students, and undergraduate students that did much of the heavy lifting when it came to collecting data and performing *in vivo* experiments. It is also necessary to acknowledge the cost of this research in terms of animal lives. While this work required the efforts of many individual people, it also required the ultimate sacrifice from a number of animals, to whom I am eternally grateful. Outside of the lab I have also enjoyed the support from my family – who was always there when I needed them – and friends, many with fur or feathers, that made each day a bit brighter.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

ADMETS	Asynchronous distributed micro-electrode stimulation
AED	Anti-epilepsy drug
ANOVA	Analysis of variance
ANT	Anterior nucleus of the thalamus
AUC	Area under the curve
BaO	Bayesian optimization
CA1	Pyramidal cell layer 1
CA3	Pyramidal cell layer 3
CEM	Cross-entropy method
cEP	Cortical evoked potential
CNR	Contrast-to-noise ratio
DA	Discrimination area
DAQ	Data acquisition system
DBS	Deep brain stimulation
DG	Dentate gyrus
DS	Discriminant score
ECoG	Electrocorticography
EHVI	Expected hypervolume improvement
EI	Expected improvement
EMG	Electromyography
FDA	Food and Drug Administration
GCL	Granule cell layer
GFP	Green fluorescent protein

GPi	Internal segment of the globus pallidus
H	Hilus
IPG	Internal pulse generator
LED	Light emitting diode
LFP	Local field potential
LS	Lateral septum
MEA	Microelectrode array
mEP	Motor evoked potential
MS	Medial septum
NHP	Non-human primate
NMSE	Normalized mean squared error
PD	Parkinson's disease
PI	Predicted improvement
RNS	Responsive neurostimulator
SANTE	Stimulation of the anterior nucleus of the thalamus for epilepsy
SDBO	State-dependent Bayesian Optimization
SOR	Spatial object recognition task
SPSA	Simultaneous perturbation stochastic approximation
SR	Stratum radiatum
STN	Subthalamic nucleus
tACS	Transcranial alternating current stimulation
TLE	Temporal lobe epilepsy
TMS	Transcranial magnetic stimulation
UCB	Upper confidence bound

SUMMARY

Neural modulation has become a fundamental tool for understanding and treating neurological and psychiatric diseases and disorders including Parkinson's disease and other movement disorders, epilepsy, depression, and a growing number of other indications. Neural modulation encompasses a range of technologies for stimulating the brain using an exogenous signal to modulate neural activity and ultimately, behavior. Electrical and magnetic stimulation are used therapeutically and have been used in research applications along with optogenetic, magnetic, and ultrasonic stimulation. The stimulation is typically defined by a set of parameters that can be adjusted to change the shape of the signal. However, delivering the desired therapy or intervention is dependent on using the correct stimulation parameters, which can be specific to the neural circuit being modulated, the disease being treated, and critically, the individual subject or patient. Researchers and medical device manufacturers are continually developing tools with unprecedented flexibility for precisely tailoring stimulation parameters for subject-specific interventions. As a consequence, this increased flexibility makes it exponentially more difficult to identify the optimal stimulation parameters for an individual. Data-driven optimization is a well-developed field of engineering that has recently emerged as a solution to the problem of selecting stimulation parameters for neural modulation. However, data-driven optimization systems must be carefully designed for each specific neural modulation problem or they risk failing to find optimal, or even reasonable, stimulation parameters. In this dissertation, I demonstrate the importance of and develop a framework for designing effective data-driven optimization solutions for specific neural modulation problems. I

develop this framework in the context of maximizing an electrophysiological biomarker using optogenetic stimulation in a rodent model. I then extend the framework to neural modulation problems in rodents and humans with relevant design goals: optimization to a setpoint, optimization of a state-dependent response, safe optimization, and optimization of multiple objectives simultaneously. Through these examples, I demonstrate how this framework can improve neural modulation by allowing for more precise subject-specific therapies and interventions, and provide general guidelines for implementing this approach.

CHAPTER 1. INTRODUCTION

Neural modulation has recently emerged as a promising therapy for the treatment of neurological and psychiatric disorders, and has been a fundamental tool for studying the circuitry and physiology of the brain. Neural modulation includes a range of therapies including, but not limited to, deep brain stimulation (DBS), transcranial magnetic stimulation (TMS) and optogenetic stimulation. DBS is currently used for the treatment of Parkinson's disease (PD) [1], essential tremor [2] for the treatment of medication resistant epilepsy [3], [4], and more recently, for treatment-resistant depression [5], obsessive compulsive disorder [6], and Tourette's syndrome [7]. For research purposes, electrical stimulation has been useful for the functional mapping of neural circuits, while optogenetics has allowed for the cell type-specific interrogation of the neural circuitry [8]. Additionally, non-invasive approaches including TMS [9] and transcranial alternating current stimulation (tACS) [10] have been invaluable for studying neurophysiology in human subjects. The common feature is that each of these technologies allows for an exogenous signal, be it electrical, optical, or magnetic, to be delivered to the brain to drive neural activity in a targeted way. The stimulation is typically defined by a set of parameters that can be adjusted to change the shape of the signal. However, delivering the desired therapy or intervention is dependent on using the correct stimulation parameters, which can be specific to the neural circuit being modulated, the disease being treated, and critically, the individual subject or patient.

1.1 Neural modulation for the treatment of neurological and psychiatric disorders

1.1.1 Neural modulation for Parkinson's disease

PD is a movement disorder characterized by the death of dopaminergic cells in the substantia nigra, which can result in motor symptoms including, but not limited to, rigidity, akinesia, bradykinesia, and tremor [11]. One of the most effective pharmaceutical therapies is levodopa, a dopamine precursor. However, patients can develop side-effects and/or a tolerance to the medication that outweighs its therapeutic benefits. When the disease is not adequately controlled with medication, DBS is considered.

In DBS for PD, electrodes with multiple contacts are implanted into one or both of the subthalamic nucleus (STN) or the internal segment of the globus pallidus (GPi) – specific nuclei of the basal ganglia. The electrode is then connected to a stimulating device (internal pulse generator; IPG) implanted in the chest that can deliver an electrical signal, referred to as a stimulation setting. In a conventional DBS device (Medtronic Activa, Abbott Infinity, Boston Scientific Vercise), the stimulation setting is a charge-balanced, biphasic, square wave defined by parameters including the amplitude and widths of each pulse, the frequency or rate of the pulses, and contact(s) on the electrode used to deliver the stimulation (**Figure 1.1**). Stimulation can be monopolar where a contact on the electrode is the cathode and the IPG is the anode, or bipolar where the anode is another contact on the electrode.

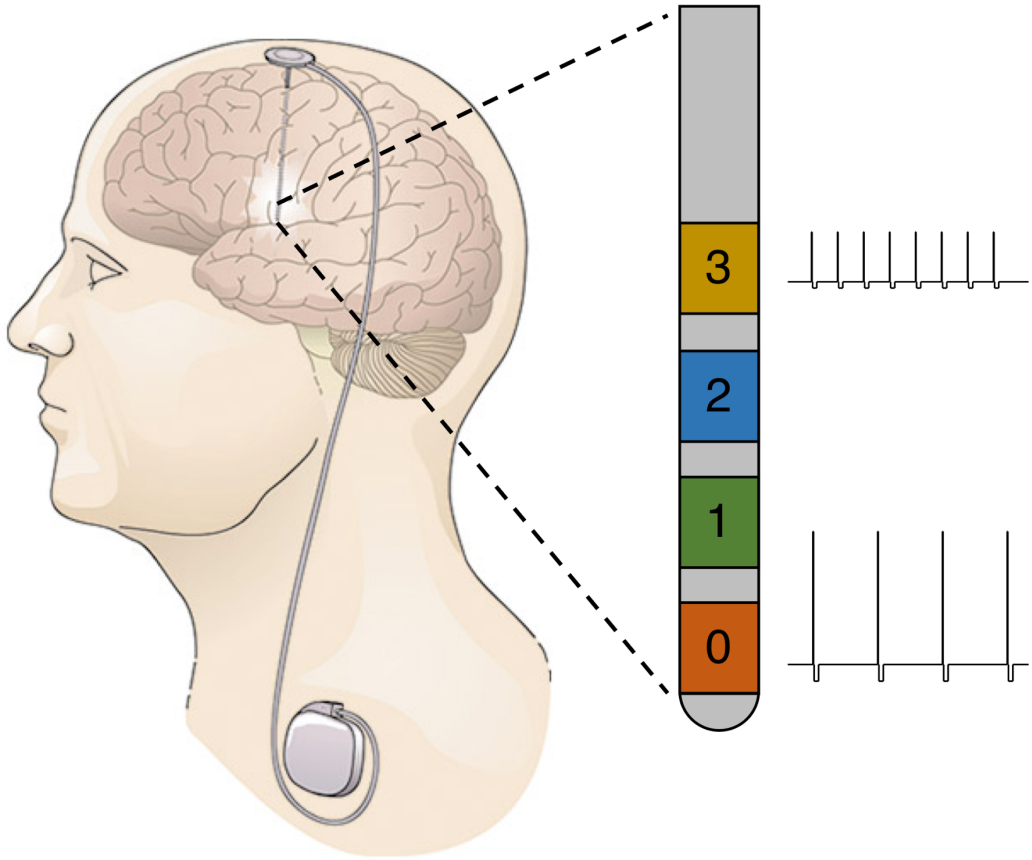

Figure 1.1: Placement of the IPG and electrode with example stimulation parameters. (*Left*) Anatomical position of the electrode for STN stimulation connected to the subcutaneous IPG in the upper chest area. (*Right*) A standard stimulating electrode with four contacts that form rings around the lead. Each contact can be configured to deliver a stimulation waveform defined by a set of parameters.

After surgery, a clinician adjusts the stimulation parameters to find the setting that works best for the individual patient. This is typically approached using a brute-force trial-and-error approach [12]. Starting with a monopolar review, the neurologist goes through each contact and increases the amplitude while observing the patient's symptoms and side-effects. If a suitable stimulation setting cannot be found (lack of symptom control, presence

of side-effects), then bipolar configurations are evaluated. To streamline this process, different guidelines and heuristics have been proposed which have the potential to improve the overall therapy [13].

Since DBS was approved by the Federal Drug Administration (FDA) for PD in 2001, over 160,000 patients have been implanted with DBS devices [14]. Over multiple randomized controlled trials, DBS of the STN or the GPi along with medical therapy has been shown to control PD motor symptoms better than medication alone [15]. More recently, ongoing research has led to the routine use of 8-contact segmented electrodes [16], and the development of more experimental closed-loop stimulation systems [17]. While these new technologies allow for more precise and effective therapies, there are several barriers to fully taking advantage of them.

The central challenge of DBS programming for PD is the size of the parameter space. Along with the device itself, the eight contacts of a segmented electrode can each be set as a cathode, an anode, or off. This allows for a total of 3^9 or 19,683 contact configurations. When additional stimulation parameters are included (e.g., amplitude and frequency), the size of the potential parameter space grows exponentially larger. This challenge is only made greater by the increasingly flexible devices being developed that offer more precise therapies, but at the cost of a more complex set of stimulation parameters [18]. Further compounding this challenge is the time necessary to measure the effects of a single stimulation setting. When adjusting stimulation settings for PD, it can take several minutes before the patient responds [19]. With only so much time to program the device, brute-force approaches quickly become intractable for larger parameter spaces [20].

Finally, as with any biological system, there is measurement noise. As the patient's underlying state (medication levels, arousal, stress) changes, identical stimulation settings can produce slightly different effects. While DBS has been and will likely continue to be an effective therapy for PD, current programming approaches may not adequately take advantage of the flexibility offered by future generations of DBS devices.

1.1.2 Neural modulation for epilepsy

Neural modulation in the form of deep brain and cortical stimulation is also now approved for the treatment of epilepsy. There have been two pivotal clinical trials of brain stimulation for epilepsy. The stimulation of the anterior nucleus of the thalamus (SANTE) trial, and the trial for the responsive neurostimulation (RNS) device developed by Neuropace.

The SANTE trial was a prospective, randomized, double-blinded, parallel group clinical study for the treatment of temporal lobe epilepsy (TLE) [3]. Bilateral stimulation of the anterior thalamic nuclei resulted in a median seizure reduction of 40.4% after 3 months of stimulation versus a median seizure reduction of 14.5% in no stimulation controls that received only anti-epileptic drugs (AED) [21]. The stimulation parameters used were 5 V, 90 μs pulses, 145 Hz, "ON" 1 min and "OFF" 5 mins. During the open label period (4 – 13 months) all participants (N = 110) received active stimulation, up to 7.5 V and 185 Hz . After the open label period, the clinicians were allowed to set any parameter provided it did not cause adverse effects. Long-term follow up showed a median 69% reduction in seizure frequency at the end of 5 years [3].

The other FDA approved brain stimulation device for epilepsy, the RNS, detects epileptic activity through electrocorticography (ECoG) and provides brief pulse of stimulation in response to the detection of a seizure. The system uses a cortical strip settled on the surface of the temporal lobe for ECoG recording and a depth electrode that provides stimulation to the hippocampus [4]. A long-term treatment study looking at the efficacy of RNS system showed seizure reduction of 75% after 9 years of treatment [4], [22]. Other clinical studies have shown the efficacy of hippocampal stimulation in suppressing seizures in patients with refractory TLE [23].

While these studies have shown the promise of using brain stimulation therapy for medication resistant epilepsy, none have achieved the ultimate goal of the therapy – seizure freedom. Just as in DBS for PD, one of the main challenges is the selection of stimulation settings. While less complex 4-contact electrodes are more commonly used, a responsive stimulation system like the RNS requires seizure detection parameters to also be selected, which creates an additional layer of complexity. Even more so than in DBS for PD, the time necessary to measure the effect of a stimulation setting poses a significant barrier. As patients do not display symptoms between seizures, a single stimulation setting is evaluated based on the number of self-reported seizures between clinic visits. As a result, the relationship between stimulation settings and therapeutic efficacy is less clear than in DBS for PD and movement disorders. This further motivates the need for approaches to efficiently search through the space of possible stimulation settings.

1.1.3 Other indications for neural modulation therapy

Beyond Parkinson's disease and epilepsy, brain stimulation therapy has found traction in the treatment of an even broader list of neurological and psychiatric diseases and disorders. DBS has been shown to be effective in other movement disorders including essential tremor [2] and dystonia [24], and was FDA approved in 1995 and 2003, respectively. More recently, DBS of the subcallosal cingulate – a white matter junction in the prefrontal cortex – has been shown to be effective for treatment-resistant depression [25]. Additional diseases and disorders for which DBS is being researched include obsessive compulsive disorder [6], Tourette's [7], and post-stroke pain [26]. While a major focus of these studies is selecting the correct anatomical target and establishing efficacy, there is no reason why these therapies would be exempt from the need for subject-specific stimulation settings.

1.2 Determining stimulation parameters for neural modulation

1.2.1 Challenges

Each of the applications of DBS for the treatment of neurological and psychiatric disorders shares a common set of challenges. First, as well-established in movement disorders, particularly PD, the efficacy of the therapy is directly tied to the stimulation setting specific to each subject. Second, unlike more common medical therapies which have dosage as the main degree of freedom, DBS has to be programmed with a stimulation setting selected from a high-dimensional parameter space that is only getting larger as more sophisticated devices are being developed. As additional stimulation parameters are added,

the size of the parameter space grows exponentially making the problem of selecting the optimal stimulation setting even more difficult, if not impossible. This property is known as the curse of dimensionality (**Figure 1.2**) [20]. Not only is it necessary to identify the optimal stimulation setting from many options, evaluating the effect of each stimulation setting can be time-consuming, particularly in DBS for epilepsy and depression. Finally, as with any biological system, the effect of a stimulation setting will have some measurement noise, and many clinical and behavioural responses will be subjective. These challenges coalesce to make it difficult to efficiently optimize stimulation settings in the current generation of DBS devices, and intractable in the next.

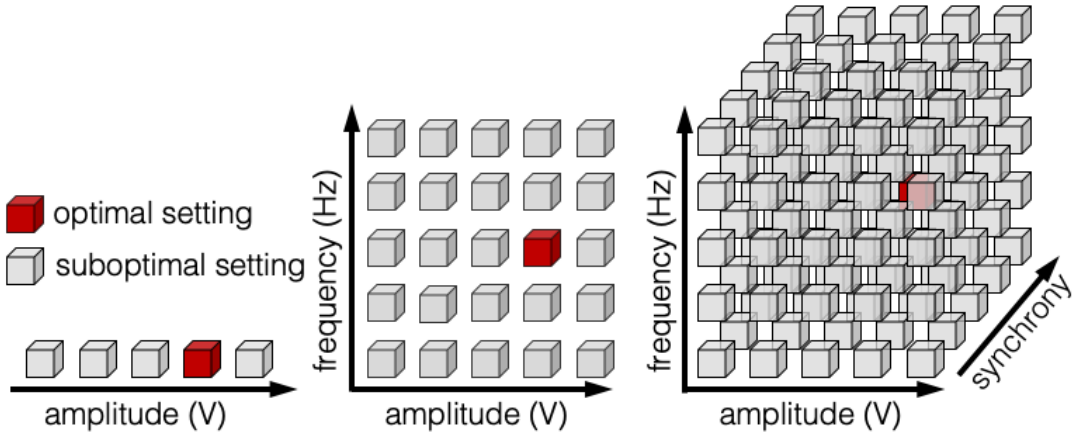

Figure 1.2: Curse of dimensionality in deep brain stimulation parameter selection.

A schematic representation of how the curse of dimensionality makes the size of the parameter space increase exponentially as new parameters are added. Each cube represents a single stimulation setting in a discretized parameter space. The red cube is the optimal stimulation setting within that parameter space, while the grey cubes are all suboptimal. Each parameter is a dimension of the parameter space. Five arbitrary values are selected for each parameter such that (*Left*) the 1D parameter space has five possible settings, (*Center)* the 2D parameter space has 25 possible settings, and (*Right*) the 3D parameter space has 125 possible settings. As a percentage of the total parameter space, the optimal setting represents 20% of the 1D space, 4% of the 2D space, and 0.8% of the 3D space.

1.2.2 Current approaches

There are a number of established approaches for selecting stimulation settings from a defined parameter space. While the approaches can vary between application, they are most developed in DBS for PD. One of the approaches to DBS programming for PD patients uses a brute-force approach to map the parameter space, and then refines stimulation according a simple line search algorithm [12]. While typically performed by a clinician observing the patient, recent work has automated this approach by using a wrist accelerometer to quantify the effect of stimulation settings in real-time [27]. This hierarchical brute-force approach has also been recently extended to 8 contact segmented electrodes [28]. A more sophisticated approach described in [13] provides a flowchart for adjusting the stimulation parameters based on specific symptoms and side-effects observed during programming as well as the neural structure being stimulated.

Beyond the more established approaches, there are multiple lines of research to address the problem of selecting stimulation settings from larger and more complex parameter spaces. One line of research has been to use biophysical computational models of the neural circuitry to evaluate the effect of stimulation settings. For PD, an Izhikevich neuron model of the thalamocortical-basal ganglia circuit was used to design stimulation settings with complex temporal patterns that were later validated in animal studies and in patients [29], [30]. While biophysical models like these and others [31], [32] can be used to improve brain stimulation, and are critical to understanding the underlying mechanisms, they are often specific to a particular neural circuit, (patho)physiology, and stimulation modality. This can make it challenging to generalize across neural modulation problems.

Another approach used a linear time invariant network model to design optimal control policies [33]. This is a more flexible approach as it can be fit to individual patients based on imaging. While powerful, optimal control approaches may be challenging to implement clinically and may face a longer translational pipeline. Finally, one subject-specific approach increasingly being used is finite-element modeling of the neural tissue to predict the spread and electrical properties of the stimulation [34].

1.2.3 *Data-driven optimization*

Data-driven optimization through direct interaction is a well-developed field of engineering that has recently gained traction as an approach to neural modulation. A data-driven optimization problem is defined by an objective function (a quantitative measure of the response to a stimulation setting; for example, the performance on a behavioral assay or the value of an electrophysiological feature), and a search space (the set of all possible stimulation settings). Once framed as an optimization problem, there are a range of established algorithms that can be applied to quickly and reliably find the stimulation setting that maximize the electrophysiological or behavioral objective. Data-driven optimization offers several advantages for selecting stimulation settings, including:

- Standardizing DBS programming across patients
- Navigating high-dimensional and noisy parameter spaces that are difficult to visualize
- When properly validated, can automate the selection of stimulation settings

- Mathematical guarantee of performance (better stimulation settings) and efficiency (in fewer samples).

Data-driven optimization for brain stimulation is based on three main assumptions. First, every subject will have a different response to a given stimulation setting. For example, depending on the positioning of the electrode a different amplitude may be necessary to stimulate the desired structure in the brain. The second assumption is that the response of a given subject – and therefore the best stimulation setting – is *a priori* unknown (**Figure 1.3**, *Left*). The goal, therefore, of data-driven optimization is to sequentially choose stimulation settings so as to find the one that maximizes the therapeutic response in as few samples as possible (**Figure 1.3**, *Right*). Finally, the third assumption is that the response to stimulation settings is *structured* in that stimulation settings near each other do not have completely random responses. This allows for the effect of tested stimulation settings to help estimate or predict the effect of untested stimulation settings when selecting which setting to choose next.

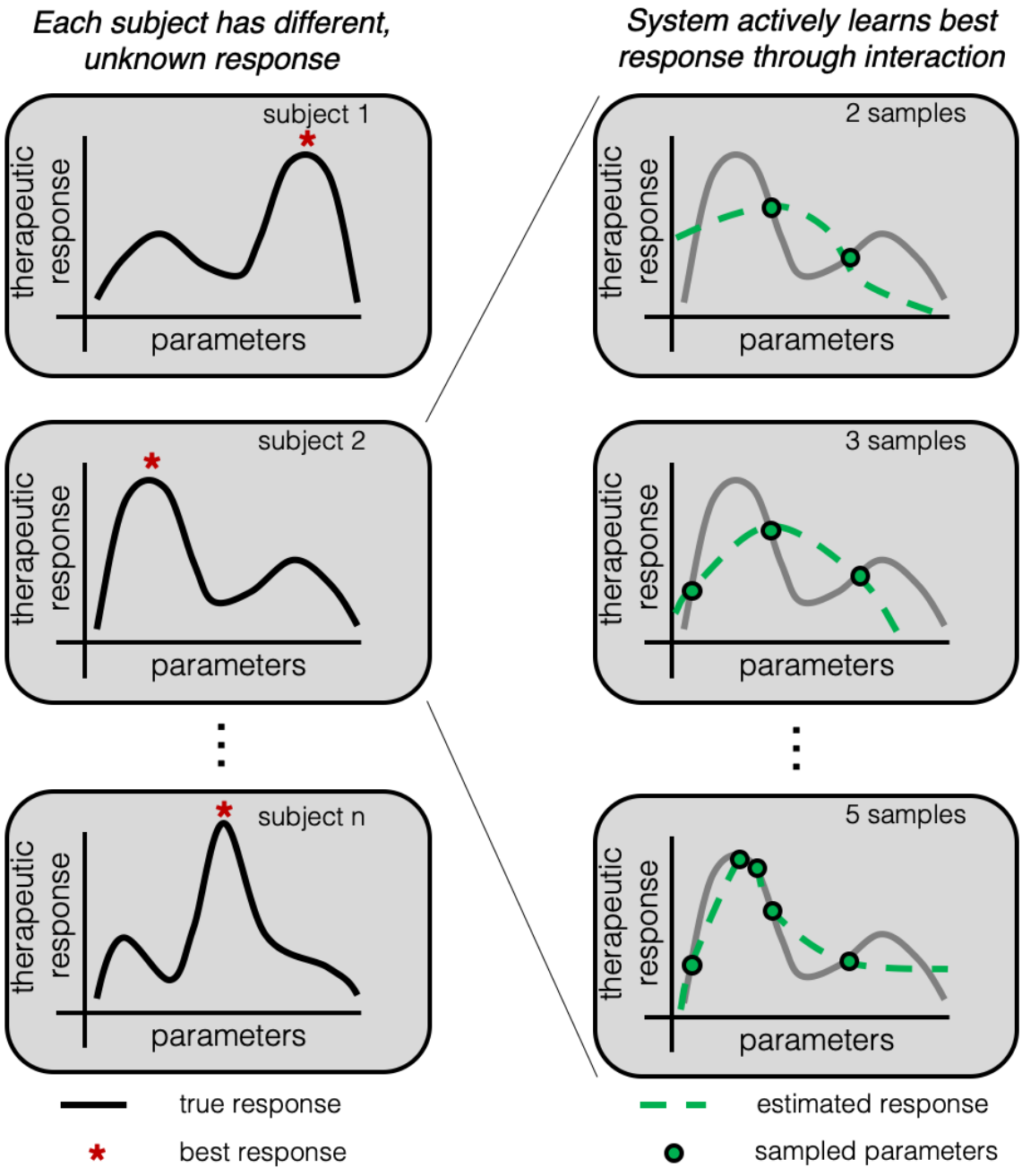

Figure 1.3: Data-driven optimization for neural modulation.

Left, each panel represents a generic space of parameters and the corresponding response to stimulation. While each subject has an underlying response to stimulation, this is not known *a priori*. *Right*, to find the optimum of an unknown objective function, a data-driven optimization algorithm carefully selects stimulation parameters and measures the response until convergence criteria are met.

Data-driven optimization started making inroads in neuroscience as a technique for designing audio-visual and audio stimuli for imaging and psychophysical studies. In [35], gradient approximation and Bayesian optimization approaches were used to identify the parameterized audio-visual stimuli that selectively activated the temporal and occipital cortices as measured with functional magnetic resonance imaging. Similarly, a variation of optimization called data-driven active learning was successfully used to efficiently learn subject-specific audiograms – a clinical test that measures a patient's auditory detection threshold across a range of frequencies [36]. For neural modulation, data-driven optimization has been applied *in silico* to find the transcranial alternating current (tACS) parameters that maximized visual phosphenes across a set of subject derived statistical models. *In silico*, data-driven optimization was used to find the linear control parameters that best controlled DBS in a mean-field model of PD, and to find the stimulation parameters that maximized a statistical model constructed from patient tremor data. However, application of data-driven optimization for neural modulation *in vivo* has been limited. One possible barrier is the need for using the correct data-driven optimization algorithm for the specific neural modulation problem, and the consequences of using an algorithm less-suited to the task.

While data-driven optimization is a potentially powerful tool for selecting stimulation settings, given the unique requirements of every different neural modulation problem, bespoke data-driven optimization algorithms must be carefully designed, or they risk failing to find the optimal, or even reasonable, stimulation settings. In this dissertation, I establish the importance of selecting the correct data-driven optimization algorithm,

present a framework for designing effective data-driven optimization algorithms for specific neural modulation problems, and demonstrate how it can be applied through several use-cases described in the following overview.

1.3 Overview

This dissertation is divided into six chapters including the introduction. In Chapter 2, I develop the foundation of the design framework in an experimentally tractable optogenetic stimulation model. In Chapters 3-6, I demonstrate how this framework can be extended and applied to several additional neural modulation problems, each with unique constraints and challenges. The experimental models used in these chapters are not intended to provide a specific therapy or study a particular physiological mechanism, but to highlight the breadth of problems that can be solved through data-driven optimization, and provide guidelines for how it can be adapted. The following is an extended summary of this dissertation highlighting the main contribution and results from each chapter.

Chapter 2. Framework development with optogenetic stimulation to maximize hippocampal gamma power

I developed this framework in the context of a rodent optogenetic system where pulsatile 465 nm light is used to stimulate the medial septum while recording downstream activity in the hippocampus. The goal was to design an optimization system to automatically search for the stimulation parameters – amplitude, frequency, and pulse width – that maximize hippocampal gamma power through direct interaction. The design framework followed four steps: 1) Data was collected from an initial cohort of subjects

representing the diversity in the response to stimulation. 2) Individual *in silico* models representing each subject in the initial cohort were constructed that estimate how a given combination of stimulation parameters affect hippocampal gamma power. 3) These models were then incorporated into a simulation platform to rapidly prototype and evaluate the performance of different data-driven optimization algorithms. 4) The algorithm with the best performance was then deployed for real-time *in vivo* experiments to search *de novo* for the stimulation pattern that maximizes hippocampal gamma power. While a relatively trivial optimization problem, this provided a controlled environment for developing the design framework. The goal was to solve a trivial problem (gamma maximization) in a non-trivial way with broader implications – designing a bespoke algorithm to find the optimum rapidly and efficiently. This work was accepted for publication in the *Journal of Neural Engineering* [37]. This chapter lays the foundation for the more significant optimization problems addressed using this design framework in Chapters 3-6.

Chapter 3. Application of the framework for setpoint objectives

It may not always be necessary, or even safe to find the stimulation pattern that maximally modulates a biomarker. Instead, it may be more appropriate to modulate the biomarker to a specific level. To accomplish this, I reframed the optimization problem to identify the stimulation parameters that regulate hippocampal gamma to a desired setpoint. Then, following the same design framework, I prototype the algorithm with the simulation platform, and then deploy it in a second set of real-time *in vivo* experiments. This work has been completed and is part of the manuscript described in the previous section.

Chapter 4. Application of the framework for state-dependent objectives

A common concern with neural modulation is that the underlying neural state may influence the effect of stimulation. This is the case with optogenetic stimulation under isoflurane anesthesia. As isoflurane decreases the conductance of K^+ channels, it restricts the range of possible stimulation frequencies. Using this as a model system, I extend the design framework for prototyping optimization approaches that learn state-dependent stimulation policies. This work has been published as a conference paper in the *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society EMBS*, 2019.

Chapter 5. Application of the framework for learnable safety constraints

Neural modulation is not without risks. Using the wrong stimulation pattern can cause uncomfortable side-effects, or even seizures in some subjects. Unfortunately, a stimulation pattern that may be optimal for one subject may be unsafe in another. In our recent experiments to maximize hippocampal stimulation amplitude, we have found that stimulation patterns that are well-tolerated by some subjects can cause memory deficits and seizures in others. Therefore, the challenge is to learn the optimal stimulation without inducing side-effects in the process. To address this, I adapted the framework to use a recent development in Bayesian optimization to incorporate online learning of safety constraints. I then used the framework to design an optimization system to safely learn the optimal subject-specific stimulation voltage. This work has been published as a conference paper in the *Proceedings of the Annual International Conference of the IEEE Engineering*

in Medicine and Biology Society EMBS, 2018, and a full journal article is being prepared for submission.

Chapter 6. Application of the framework for multiple objectives

In many applications, there can be multiple, different objectives that need to be optimized simultaneously. This is the case in stimulation of the STN for PD. In this therapy, it is necessary to use sufficient current to stimulate the STN and produce a therapeutic effect. However, over-activation of an adjacent structure, like the internal capsule, can cause side-effects such as involuntary muscle spasms. This creates two mutually competitive objectives for which there is no one optimal stimulation pattern. Instead, there is a set of stimulation patterns that describe the optimal trade-off between these two objectives, known as the Pareto set. Using data collected from intra-operative stimulation experiments, I used the framework to demonstrate how multi-objective surrogate model optimization can be used to efficiently learn subject-specific Pareto set of stimulation patterns. This work has been submitted to *Journal of Neural Engineering*.

Chapter 7. Conclusions

To conclude, the steps of the framework are reviewed by discussing how each was implemented across the different neural modulation problems, considerations for applying the steps to a new neural modulation problem, and more advanced approaches that may be incorporated in future version of this framework. Finally, we consider the engineering, basic science, and clinical perspectives of the framework.

CHAPTER 2. FRAMEWORK DEVELOPMENT IN AN OPTOGENETIC STIMULATION MODEL

2.1 Introduction

We develop this framework in the context of a rodent optogenetic system where pulsatile 465nm light is used to stimulate the medial septum while recording downstream activity in the hippocampus [38], [39]. The goal is to design a data-driven optimization system that looks for the combination of optogenetic stimulation parameters – amplitude, frequency, and pulse width – that maximizes the magnitude of gamma power (33-50 Hz) measured from the hippocampal local field potential (LFP). While a relatively trivial optimization problem, this model provided a controlled environment for developing the design framework that captured many of the challenges of neural modulation, including the time necessary for each iteration and the sample-to-sample variability in the response to stimulation. The goal was to solve a trivial problem (gamma maximization) in a non-trivial way with broader implications – developing a bespoke algorithm by using a design process that can be replicated for other neural modulation problems.

2.2 Methods

This framework follows four steps shown in **Figure 2.1**: 1) We start by exhaustively sampling a range of stimulation parameters from an initial cohort of multiple subjects. 2) We then use this data to construct an *in silico* model for each subject that estimates how a given combination of stimulation parameters affect hippocampal gamma

power. 3) These models are then incorporated into a simulation platform to evaluate the performance of different data-driven optimization algorithms. 4) The algorithm with the best performance is then deployed *in vivo* in new subjects to search for the stimulation pattern that maximizes hippocampal gamma power (**Figure 2.1C**). We then extend this algorithm to identify the stimulation parameters that regulate hippocampal gamma to a desired setpoint. Following the same framework, we prototype the algorithm with the simulation platform, and then deploy it in a second set of real-time *in vivo* experiments.

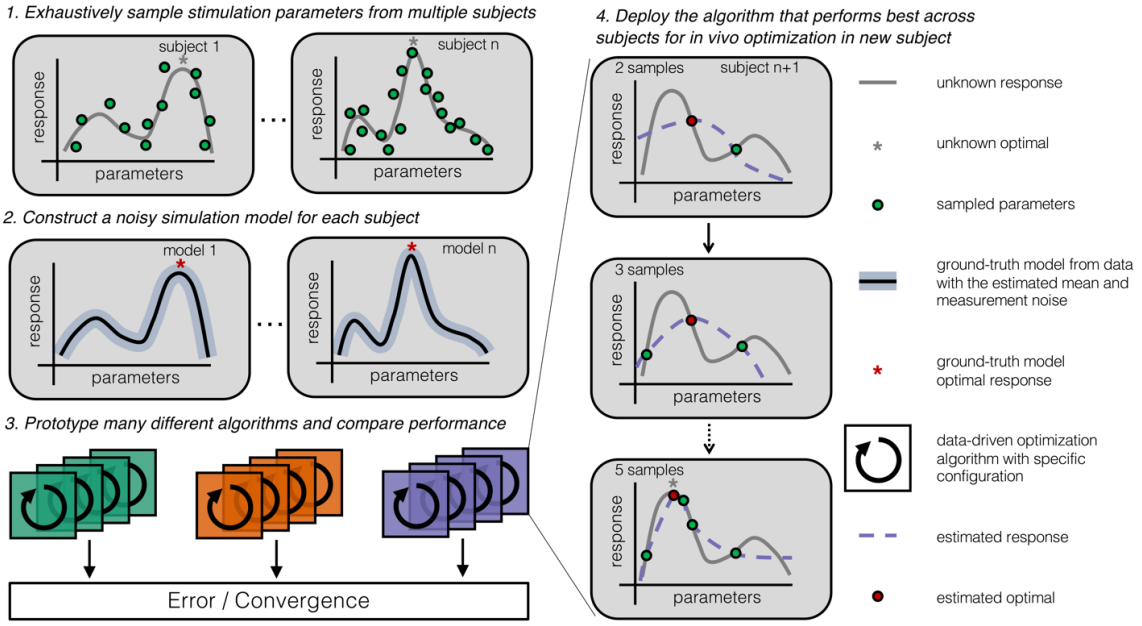

Figure 2.1: Conceptual overview of optimization design framework.

The design process for building a data-driven optimization system. In this conceptual overview, we consider a generic space of (stimulation) parameters that have an effect on a response of interest. The relationship between the parameters and the response is assumed to vary across subjects, have appreciable measurement noise, and be *a priori* unknown. The process of designing and validating an optimization algorithm follows four steps. 1) Pilot data is collected by exhaustively sampling stimulation parameters from an initial cohort of multiple subjects. 2) This pilot data is then used to construct noisy simulation models for each individual subject that can serve as proxies for their *in vivo* counterpart. 3) Many different data-driven optimizations algorithms are evaluated by rapidly testing their

performance on the proxy models. 4) The data-driven algorithm that performs the best is then deployed *in vivo* in a new subject that was not exhaustively sampled. This is a generalizable process that can be applied to design data-driven optimization algorithms for other neural modulation contexts.

2.2.1 Surgery, virus injection, implant

A total of 6 adult male Sprague-Dawley rats (2-3 month old; 250-300 g) from Charles River Laboratories (Wilmington, MA, USA) were used for the study. All animals were maintained within a 12/12 light/dark cycle vivarium with unlimited access to food and water. All procedures were conducted in accordance with Emory University's Institute for Animal Care and Use Committee.

For each subject, two surgical procedures under anesthesia (1.5-4% inhaled isoflurane) were conducted as previously described [40]. First, the viral vector (AAV5-hSynapsin-Channelrhodopsin2-eYFP) was injected into the medial septum just to the right of the midline at a 20° angle to the dorsal-ventral axis (0.40 mm anterior, 2.12 mm lateral at the 20° angle, 5.80 mm ventral to pia along the rotated axis). A volume of 1.8 µL containing 10^{12} particles/mL was injected with a rate of 0.35µL/minute using a pulled-glass pipette attached to a stereotactically mounted injector (Nanoject, Drummond Scientific Co., Broomall, PA, USA). Once the pipette was withdrawn, the scalp was stapled closed, and Meloxicam was administered as an analgesic (3-5 mg/kg).

The second survival surgery was conducted after two weeks, allowing time for recovery and optogenetic channel expression. A 16-channel multielectrode array (MEA;

Tucker Davis Technologies (TDT), Alachua, FL., USA) targeting hippocampal CA3 and CA1 was implanted in the right dorsal hippocampus (centered at 3.50 mm posterior and 2.80mm lateral to bregma). The MEA was lowered ventrally into the brain until single unit activity was observed from both the CA1 and CA3 regions. The ferrule was then driven into the reopened original injection craniectomy at a 20° angle to the dorsal-ventral axis to approximately 5.8mm from the pia along the rotated axis. Correct ferrule depth was determined by applying a 17Hz, 10ms, 50 mW/mm^2 stimulation for 10s. Finally, the craniectomy was sealed with dental acrylic to secure the electrode and the ferrule in place. Five 2 mm stainless steel screws were mounted on the skull for electrode's ground and reference wires as well as for the structural support.

2.2.2 Grid search experiments

After recovering from the second survival surgery, three subjects underwent several grid search experiments. Light was delivered through the implanted fiber optic at all combinations of amplitude (10, 30, and 50 mW/mm^2), pulse-width (2, 5, and 10 ms) and frequency (5, 7, 11, 17, 23, 35, and 42 Hz) for a total of 63 combinations of stimulation parameters. Stimulation parameters were applied in a random order for a duration of 20 seconds, separated by at least one minute. This experiment was repeated three times for each of the subjects. Throughout the experiment, LFPs were recorded from the hippocampus using an RZ2 BioAmp Processor and a PZ2 pre-amplifier (Tucker Davis Technologies (TDT), Alachua, FL., USA). Signals were recorded at a sampling rate of 24414 Hz, and then down sampled to 2000 Hz to limit the computational load during later real-time experiments.

2.2.3 Neural feature calculation

For each stimulation sample, the neural features representing hippocampal gamma power were calculated for the entire stimulation-on window, and then averaged across the 16 channels in the hippocampal MEA. Channels with damaged connections were excluded. The power spectral density (PSD) was estimated for the entire 20 second recording (NFFT=2^16) during stimulation using the multi-taper method from the Chronux toolbox (http://chronux.org) and then by summed from 33-50 Hz.

2.2.4 Ground truth model for simulation experiments

The stimulation parameters (amplitude, frequency, and pulse-width) and the gamma power during stimulation were then fit to a Gaussian process model mapping the parameters to the corresponding effect on hippocampal gamma power [41].

A Gaussian process is composed of a mean function:

$$\mu = m(\mathbf{x}') \tag{2.1}$$

and a covariance function:

$$\Sigma = k(\mathbf{x}, \mathbf{x}') \tag{2.2}$$

where $\mathbf{x}$ is a vector containing the data used to construct the model, and $\mathbf{x}'$ is the sample point. In our application, the inputs to the model are the relevant stimulation parameters:

amplitude, pulse-width, and frequency. The output is the gamma power, Y_{gamma}, during stimulation, drawn from a Gaussian distribution:

$$Y_{gamma} \sim \mathcal{N}(\mu, \Sigma) \tag{2.3}$$

Ultimately, this model allows for an optimization algorithm to interact with a black-box function that, for a given set of stimulation parameters, produces a stochastic response drawn from the expectation and variance learned directly from previous experiments.

2.2.5 Optimization algorithms

The simulation experiments consisted of evaluating three optimization algorithms (**Figure 2.2**), Each with 12 different configurations. Each configuration was evaluated with 30 trials for each of the three subject models, for a total of 3*12*30*3=3,240 trials. Each trial consisted of 100 samples. Simulation code was written in MATLAB.

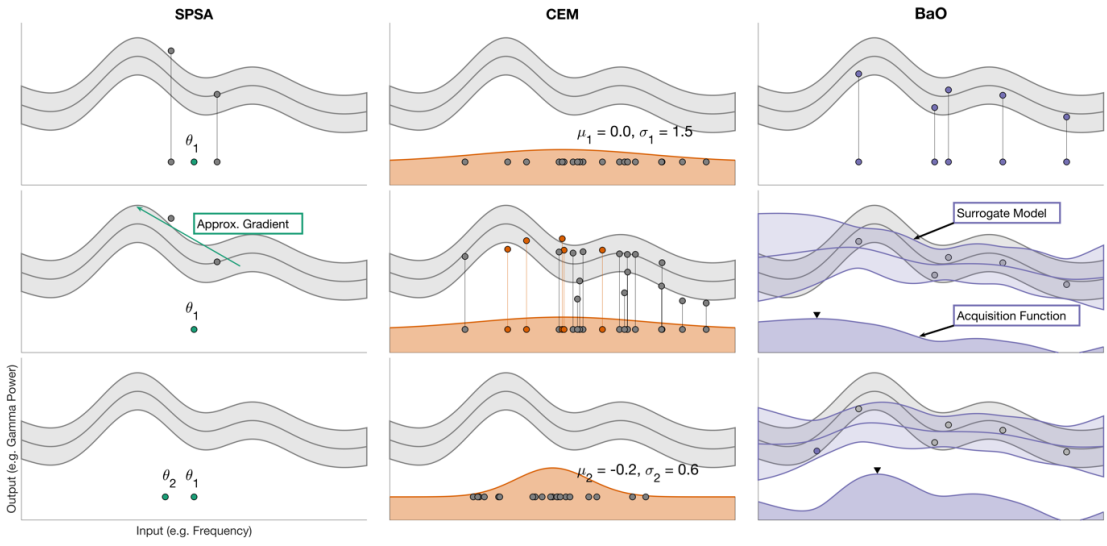

Figure 2.2: Overview of the 3 optimization algorithms.

Left panel: SPSA. *Top,* Initially, an estimate of the optimal argument (theta, green circle) is proposed. Next, the initial theta is simultaneously perturbed in multiple directions to define additional search points, (theta 1-n). *Middle,* These points are then sampled, and the gradient from these points is estimated. *Bottom*, A new value for theta along this gradient is selected and the process is repeated. *Middle panel*: CEM. *Top,* An initial distribution centered within the parameter space with a wide covariance is specified. A fixed number of sample points are selected from this distribution. *Middle,* Each of these sample points are then evaluated, and those that best satisfy the objective function are identified as *elite* samples. *Bottom,* The mean and covariance of the arguments of the elite samples are then estimated and used for the distribution in the following iteration. *Right panel:* BaO. *Top,* During an initial burn-in phase, sample points are selected from a uniform distribution and evaluated. *Middle,* A Gaussian process model is then fit to the existing data. Based on this incomplete model, an acquisition function is then used to propose a candidate sample point based on the expectation and uncertainty estimated by the model. *Bottom,* This proposed sample point is then evaluated, the model is updated with the new data, and the process is repeated.

2.2.5.1 SPSA

SPSA is a variant of gradient descent that uses several samples to calculate an approximation of the gradient [42]. An overview of the SPSA algorithm is shown in **Figure 2.2**, *Left Panel*. This allows the algorithm to effectively search objective functions with measurement noise. The goal of SPSA is to minimize a loss function $L(\theta)$ that is a scalar valued measure of performance and θ is an n-dimensional vector of parameters to be optimized. An initial estimate of the optimum θ_k is randomly drawn from a uniform distribution over the parameter space. Then SPSA generates multiple perturbations of theta to approximate the gradient:

$$g(\theta) = \frac{\partial L(\theta)}{\partial \theta} \approx \frac{y\left(\hat{\theta}_k + c_k\Delta_k\right) - \left(\hat{\theta}_k - c_k\Delta_k\right)}{2c_k} \tag{2.4}$$

where Δ_k is the standard unit vector for the k^{th} dimension of the parameter space, and c_k is the step-size of the perturbations. Based on this gradient, the estimation of the optimal theta is updated by

$$\hat{\theta}_{k+1} = \hat{\theta}_k - a_k \hat{g}(\hat{\theta}_k) \tag{2.5}$$

The SPSA gains a_k and c_k are iteratively reduced after each iteration following the schedules: $a_k = \frac{a}{(A+k+1)^\alpha}$ and $c_k = \frac{c}{(k+1)^\gamma}$, where a, A, α, c, and γ are configuration settings.

2.2.5.2 CEM

CEM is a general evolutionary algorithm suited for continuous and combinatorial multi-extrema optimization and importance sampling [43]. An overview of the CEM algorithm is shown in **Figure 2.2**, *Center Panel*. In our implementation, the CEM algorithm is initialized by specifying a Gaussian distribution is defined with a mean, μ_k, centered in the parameter space and covariance, σ_k, large enough so that it is essentially uniform. M candidate points are sampled from this distribution, and their performance ranked. The N samples with the greatest performance, Θ_{k}^{+}, are selected, and their mean and covariance are used to update the Gaussian distribution according to:

$$\mu_{k+1} = \Gamma \frac{\Sigma \Theta_{\mathrm{k}}^{+}}{\mathrm{N}} + (1 - \Gamma)\mu_k \tag{2.6}$$

$$\sigma_{k+1} = \Gamma cov(\Theta^{+}) + (1 - \Gamma)\sigma_k \quad (2.7)$$

where Γ is the learning rate.

2.2.5.3 BaO

BaO is a model-based optimization that has been used for optimization of expensive cost functions [44]. An overview of the BaO algorithm is shown in **Figure 2.2**, *Right panel*. As the BaO algorithm samples the objective-function, it iteratively builds a model of the underlying data, which is then used to select the area of the parameter space to be subsequently sampled.

In our implementation, the BaO algorithm is initialized by randomly selecting 10 burn-in samples from a uniform distribution over the parameter space. The burn-in samples are sequentially applied while the response is measured. The data is then fit to a Gaussian process model by maximizing the log marginal likelihood with regard to the hyperparameters. An acquisition function is then applied to the model to determine the next sample by considering the trade-off between regions of the parameter spaced expected to be near optimal (exploitation) and that have high uncertainty – but the potential to be optimal (exploration). The next sample is selected, applied, the model is updated with the new data, and the process is repeated.

We use a Gaussian process with a constant mean function and a third order Matérn kernel with automatic relevance detection for the covariance:

$$k(x_i, x_j) = \sigma_f^2(1 + \sqrt{3}r)\exp(-\sqrt{3}r) \tag{2.8}$$

where k: covariance, σ_f^2: variance of observation (objective function) and

$$r = \sqrt{\sum_{m=1}^{d} \frac{(x_{im} - x_{jm})^2}{\sigma_m^2}} \; where\; \sigma_m^2 \text{: } variance\; of\; x_m \tag{2.9}$$

The variables σ_f^2 and σ_m^2 and the mean, μ, are hyperparameters that are adjusted to maximize the likelihood.

In this analysis, we consider three acquisition functions:

Predicted improvement:

$$H_{PI}(\mathrm{x}) = \Phi\left(\frac{\mu(x) - f(x^+) - \xi}{\sigma(x)}\right) \tag{2.10}$$

Where x^+ is the currently estimated optimal input and ξ is a constant hyperparameter to facilitate exploration.

Expected improvement:

$$H_{EI}(x) = \begin{cases} (\mu(x) - f(x^+) - \xi)\Phi(z) + \sigma(x)\phi(z) & if\ \sigma(x) > 0 \\ 0 & if\ \sigma(x) = 0 \end{cases} \tag{2.11}$$

$$z = \begin{cases} \frac{(\mu(x) - f(x^+) - \xi)}{\sigma(x)} & if\ \sigma(x) > 0 \\ 0 & if\ \sigma(x) = 0 \end{cases} \quad (2.12)$$

Upper confidence bound:

$$H_{UCB} = \mu(x) + \sqrt{\eta \kappa_t}\sigma(x) \quad (2.13)$$

where the sequence κ_t was defined as $\kappa_t = 2\log\left(\frac{t^2\pi^2}{6}\right)$, t is the number of samples, and η is a hyperparameter that adjusts how heavily uncertainty is weighed.

2.2.6 Performance objectives

2.2.6.1 Distance from the maximum

The distance from the maximum was calculated using the expectation of the ground-truth model from the optimal stimulation parameters proposed by the optimization algorithm. In other words, whenever an optimization algorithm reaches an update (SPSA: 3 samples, BaO: 1 sample, CEM: variable), the current estimate of the parameters that maximize the objective function (x+) is used as the input to the ground-truth model to produce y+. The distance from the optimum is then calculated as the output of the true optimal parameters (y*) minus y+:

$$DM = y^* - y^+ \tag{2.14}$$

2.2.6.2 Convergence rate

To measure the overall convergence rate of an optimization algorithm/configuration, the distance from maximum as a function of samples was fit to an exponential function of the form

$$DM = ae^{bt} \tag{2.15}$$

Where DM is the distance from the max as described previously, and x is the number of samples collected thus far. This was done for each optimization trial, and the coefficient **b** was considered the convergence rate. A more negative coefficient **b** indicates that the algorithm converged to a proposed optimum more rapidly than one with a less negative coefficient **b**. It should be noted that the convergence rate and the final distance from maximum represent different aspects of the optimization performance. For example, it is possible for an algorithm to rapidly converge, but to a parameter set that is far from the maximum.

2.2.7 Simulation experiments

In the first application of the framework, the simulation models (Chapter 2, Section 2.5) were used to evaluated the performance and behavior of the data-driven optimization algorithms (Chapter 2, Section 2.6), according to final error and convergence rate. The

simulation models for each of 3 subjects (R1-R3) were applied to each of the 3 representative algorithms with 12 different configurations. Each configuration of each algorithm was evaluated for 30 trials of 100 samples per trial. In between each trial the system was reset and the optimization algorithm was started from scratch.

2.2.8 In vivo optimization

Two animals (R4 and R5), used to validate the optimization platform were prepared using the same procedure as for those used to collect exploratory data. After at least two weeks for recovery, subjects underwent the optimization experiments. The experiments were conducted in a single session consisting of two phases: optimization and validation separated by 15 minutes. During the learning phase, the optimization algorithm searched for the stimulation parameters that maximize the objective function, while the testing phase verifies that the learned parameters were reproducible and not the product of an unrelated neural state change (sleep/wake), or the consequence of repeated stimulation. The optimization and test phases were performed once per day, except for one subject that underwent two experiments within a single day. Each subject underwent four sets of optimization/validation.

2.2.8.1 Optimization phase

In the optimization phase, each subject was connected to a light emitting diode (LED; Plexon) via a fiber optic patch cable, and to the data-acquisition (DAQ) system

(Tucker Davis Technologies, Alachua FL). A custom software application (MATLAB 2014b) was used to stream the data from the DAQ and control the output of the LED. The software application was then configured to run the best performing optimization algorithm based on the findings from the simulation experiments. Each sample consisted of 20s of stimulation separated by 80s. Ten initial samples drawn from a uniform random distribution were used for the burn-in period, followed by 20 samples guided by the algorithm. For each stimulation, gamma power was calculated as described previously.

2.2.8.2 Validation phase

During the validation phase, 30 stimulations were delivered – randomized between sham stimulation, the stimulation parameters supposed to induce the highest hippocampal gamma power (estimated optimal), and the one supposed to induce the lowest gamma power (estimated worst). Each stimulation was applied as in the optimization phase (20 seconds of stimulation followed by 80 seconds of washout between samples).

2.2.9 Analysis and statistics

In the simulation experiments, final error and convergence rates were compared within subject using ANOVA and followed by Tukey post-hoc testing (MATLAB 2020a). The variability between subjects was not a focus of this study so these results were not explored quantitatively. The testing phase data was analyzed using ANOVA followed by Tukey-Kramer post-hoc testing (MATLAB 2020a) to compare the hippocampal gamma power induced by sham, estimated optimal, and estimated worst stimulation parameters.

2.3 Results

2.3.1 Ground-truth models map from medial septum optogenetic stimulation parameters to hippocampal gamma power

Hippocampal recordings during optogenetic stimulation of the medial septum were collected from three subjects as part of a previous study [38], [39]. (**Figure 2.3A**) In this study, a grid of stimulation parameters with 63 combinations of amplitude, frequency, and pulse width, was specified. From this grid, 1267 samples were collected, each sample consisting of the stimulation pattern applied and the measured hippocampal gamma power (**Figure 2.3B**,**C**). As previously observed in [39], the pulsatile stimulation produces a sharp peak in the spectral power at the stimulation frequency along with harmonics at multiples of that stimulation frequency (**Figure 2.3B**).

Using this data set, subject-specific models mapping the stimulation parameters (amplitude, frequency, and pulse width) to the hippocampal gamma power were constructed by fitting the data to a Gaussian process regression model [45]. The expectation of the Gaussian process models corresponding to each subject are shown in **Figure 2.3D**. In each of these representations, the three-dimensional model has been projected into two dimensions, omitting pulse width as the data showed it had only a slight effect on the hippocampal gamma power during stimulation (Chapter 2, Section 5.1). Plotting the expectation of each of the subject-specific models highlights the slight variation between the subjects. The primary distinguishing feature is the presence or absence of a subtle local maximum at a stimulation frequency of 17Hz and amplitude of 50 mW/mm^2. This local

maximum can be seen in subject R2, while R3 has an inflection point (i.e., is flat) at 17 Hz, but only has a single global maximum. R1 only has the single maximum at 42Hz/50 mW/mm^2. Looking at the measured data (black dots), it is clear that there is a degree of uncertainty in the response to identical stimulation parameters. Overall, the effect of stimulation on hippocampal gamma power is relatively consistent between subjects, which will allow us to assess the performance of an optimization algorithm on new animals without performing an exhaustive grid search. Models for control subjects transfected with a viral construct missing the opsin necessary for the cells to respond to optogenetic stimulation are shown in in Chapter 2, Section 5.2.

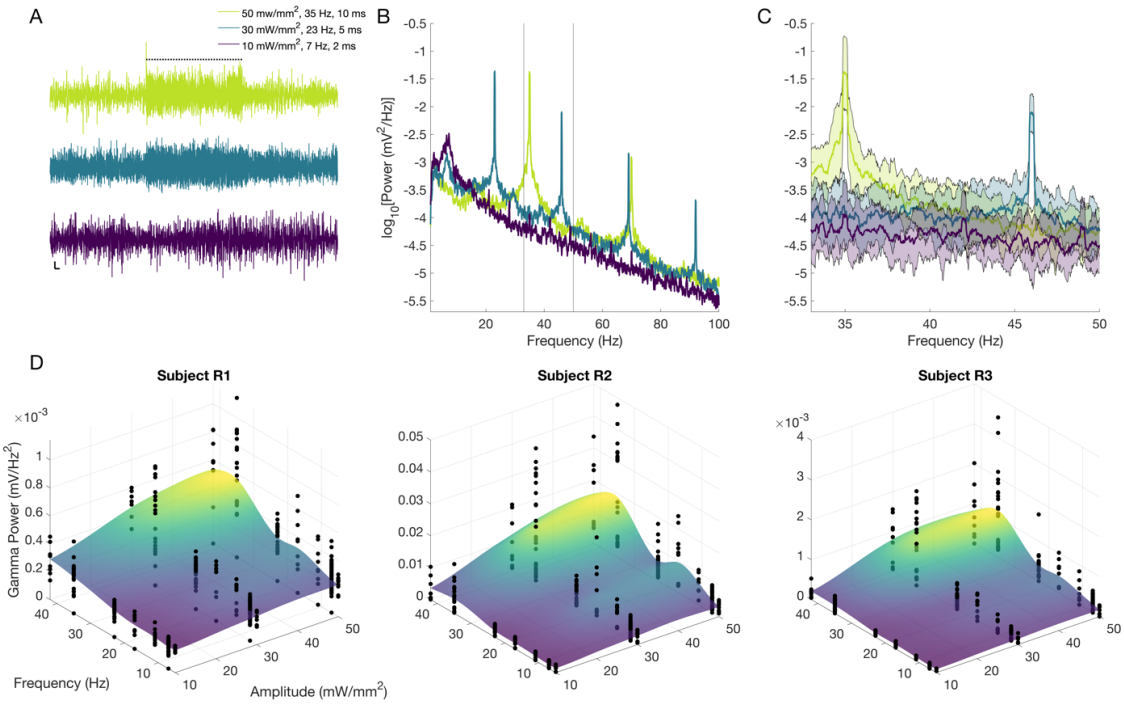

Figure 2.3: Construction of the of ground-truth objective function models.

Objective function models generated from previously collected data. A) Local field potentials from three representative samples in subject R1, scale bars for 1s and 0.1 mV. B) Log power spectral density averaged over 9 samples across 3 recording sessions.

Vertical black lines indicate the gamma frequency range (33-50 Hz). C) Focus on gamma frequency range. Shaded area indicates standard deviation. D) The surface shows the expected response to stimulation at a given frequency and amplitude. Each dot is an actual measurement from a single sample. Color gradient is provided solely for contrast and to improve visualization.

2.3.2 Platform for prototyping data-driven optimization

The models mapping medial septum stimulation parameters to hippocampal gamma power were used to construct a simulation platform for prototyping optimization algorithms. The prototyping platform was used to evaluate the three representative data-driven optimization algorithms, each with 12 different configuration settings that govern its behavior (e.g., learning rate, parameter space exploration) selected to highlight the breadth of behaviors possible with the algorithms, and based on the experimental constraints. Each configuration of each algorithm was evaluated for 30 trials of 100 samples per trial. See *Methods* for a detailed overview of the optimization algorithms and their configurations.

Figures 2.4A-C show the final performance of the optimization algorithms at each configuration for subject R1. Results for other subjects are detailed in Chapter 2. Section 5.3, but the trends were similar. Overall, for the majority of hyperparameter settings, BaO had the lowest final error. Comparing these final results, there was a difference in the sensitivity of the optimization algorithms to their configurations settings. SPSA was the most sensitive, with CEM showing some sensitivity, and BaO being sensitive to changes in configuration primarily when using the predicted improvement (PI) acquisition function – the strategy for selecting the next stimulation pattern. Notably, when configured to reduce exploration (e.g., the UCB(0.01) algorithm), BaO had a slightly larger final error

comparable to BaO with the PI acquisition function. (See *Methods* for a description of the algorithm configurations.)

Figures 2.4D-F show single-trial representative traces from each of the algorithms for the hyperparameters that produced the lowest overall final error. These traces demonstrate the difference in the search strategies that characterize each algorithm class. The SPSA algorithm (**Figure 2.4D)** shows rapid jumps in the estimate of the optimal parameters (solid green line) as new local maxima are found, while the parameters sampled (dotted green line) start with a high variance that decreases as the algorithm reduces the aggressiveness of the exploration with each subsequent sample. For CEM (**Figure 2.4E**), the estimate of the optimal parameters (solid orange line) evolves much more smoothly as each subsequent generation overlaps with the previous, while the variance of the search points (dotted orange line) also decreases as the variance of the elite samples is always lower than that of the overall population. Finally, for BaO (**Figure 2.4F**), the estimate of the optimal parameters (solid purple line) is rapidly identified for the amplitude and frequency, and more slowly for pulse-width. In contrast to SPSA and CEM, the variance of the search points (dotted purple line) increases later into the optimization process. This is due to the UCB acquisition function that increases the weight of the exploration term logarithmically with the number of samples.

In **Figure 2.4G** the overall convergence behavior across trials is shown for the representative optimizers with the lowest error. In the first 30 samples of the optimization process, BaO has the lowest error, followed by CEM then SPSA. After the 45^{th} sample SPSA has a marginally lower error than CEM. The final error for each of the three subject-

specific models is shown in **Figure 2.4H**. In all subjects, BaO has the lowest final error. In subjects R2 and R3, CEM performs nearly as well as BaO, however, in R1 CEM performs worse than both SPSA and BaO. **Figure 2.4I** compares the convergence rate of each representative optimization algorithm across subjects and shows that BaO converges most rapidly, while SPSA and CEM are significantly slower.

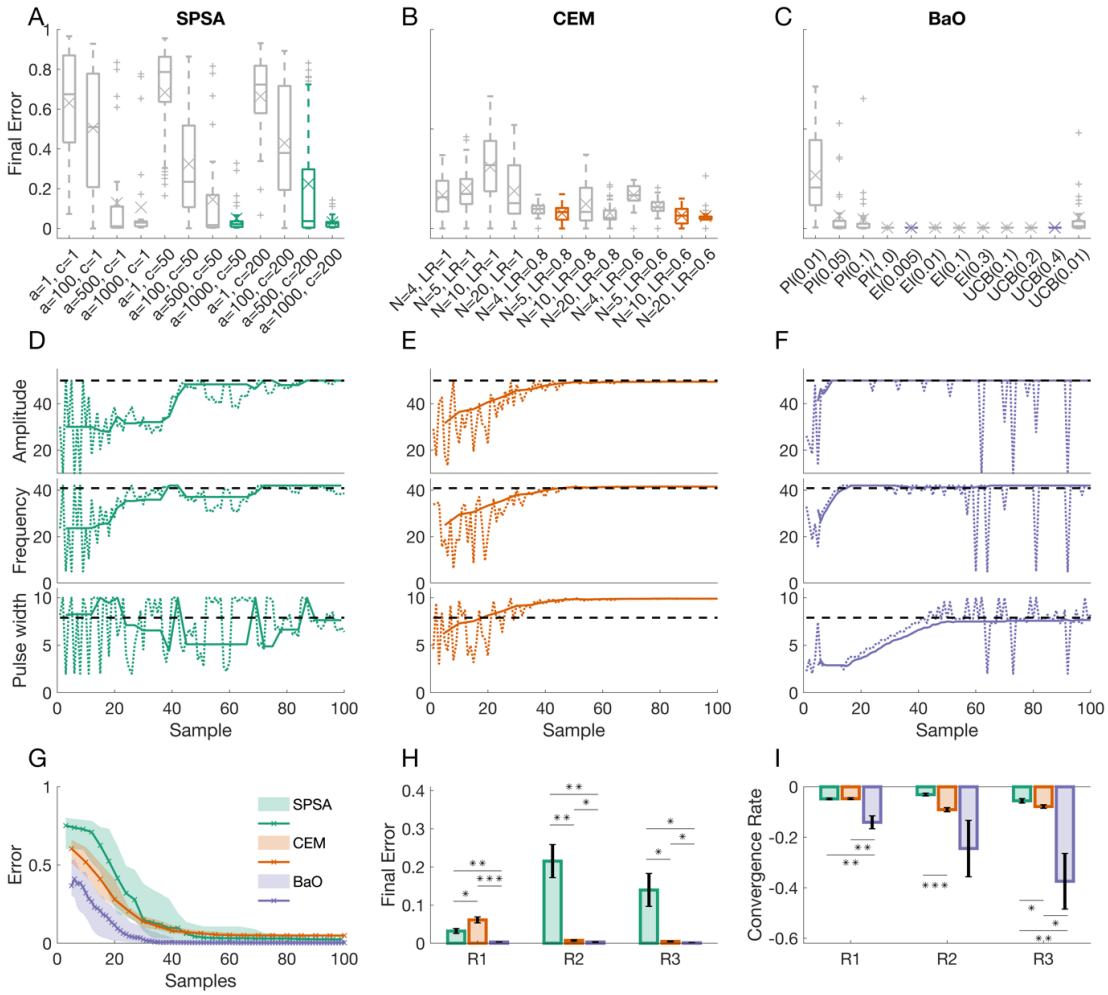

Figure 2.4: Summary of output from the prototyping platform.

(A-C) The final error statistics for running the optimization algorithms under different configurations on a single subject. Boxplots show the quartiles and the 'x' indicates the mean error. The labels on the x-axis ticks describe the specific configuration used for each optimization experiment. The highlighted results indicate that for a given optimization class, that specific configuration had the lowest final error for at least one of the subjects. (D-F) Each subplot shows a trace for a single representative trial for a single subject using the best configuration for that subject. The colored solid line indicates the stimulation parameters the optimization algorithm estimated to be optimal, while the colored dotted line shows the actual stimulation parameters sampled. Black dashed line is the ground-truth optimal stimulation parameter known from the model. (G) The overall trajectory of the final error for the class-representative configuration with the lowest error for that subject. Each trace indicates the median and quartiles across all trials at that sample number. Markers indicate the samples at which the optimal stimulation parameters could be estimated as it can vary between optimization algorithms. (H) The mean final error across all trials of the algorithm/configuration with the lowest final error for each of the subjects. Error bars show the standard error across all trials. *$p<0.01$, **$p<1x10^{-4}$, ***$p<1x10^{-7}$. (I) Same as **2.4H** for convergence rate. *$p<0.05$, **$p<0.01$, ***$p<1x10^{-5}$.

Based on these results, we can infer that the BaO algorithm using the UCB acquisition function with configuration setting nu=0.4, henceforth referred to as UCB(0.4), is best suited to the task of identifying optogenetic medial septum stimulation parameters that maximize hippocampal gamma power. While these findings support the potential utility of data-driven optimization for neural modulation, they are limited by the *in silico* nature of the experiments. We therefore sought next to validate them *in vivo*.

2.3.3 Validation of Bayesian optimization in vivo

Two subjects (R4 and R5), not included in the development of the prototyping platform, underwent optimization to maximize hippocampal gamma power using the UCB(0.4) algorithm in a prospective testing phase three times over the course of two weeks. Conditions of the experiment were analogous to those used for prototyping *in silico*. Results from real-time optimization with UCB(0.01) are presented in Chapter 2, Section 5.4.

2.3.3.1 Optimization phase

Figure 2.5 shows the trajectory of the algorithm during the optimization phase *in vivo*. At each sample, the estimated optimal stimulation parameters and the predicted hippocampal gamma are shown with solid lines. The actual stimulation parameters sampled are shown with dotted lines. In all trials, the algorithm rapidly converged on expected amplitude and frequency parameters within 10-15 samples. In contrast, the algorithm's estimate of the pulse width did not readily converge in all trials, and the final estimate of the optimal pulse widths was more varied (**Figure 2.5A and C**). The final estimated amplitude and frequency parameters strongly agree with both the modeled data and the optimization results. Moreover, the 'lack' of convergence and wide range of final values for pulse width also agrees with our previous results, and confirms pulse width's weak influence on hippocampal gamma above 2 ms. **Figure 2.5D** shows the spectrogram of a representative trial with subject R4. In the beginning of the optimization, the first 10 samples are randomly selected in what is referred to as the 'burn-in'. This can be seen from the variability in the amplified frequency bands in the spectrogram, both in the frequency range of the band and in its brightness as different combinations of amplitude and frequency are applied. In the six samples following this initial burn-in period, the amplified frequency bands become focused within the 30-42 Hz range and display more consistent brightness as the algorithm begins to converge on the optimal stimulation parameters. Finally, through the rest of the learning phase the amplified frequency bands are nearly uniform in both frequency range and brightness, as nearly identical stimulation parameters are applied by the algorithm. In comparing the individual trials within and between

subjects, there is considerable consistency in the behavior and performance of the optimization algorithm in this task. Moreover, at the end of each trial the estimated optimal stimulation parameters for amplitude and frequency fall within a very narrow range (45-50 I, and 35-42 Hz) corresponding to the back corner of the parameter space, as represented in **Figure 2.3D**. While the estimated optimal stimulation parameters are consistent between trials and subjects, the effect of the stimulation on gamma power is less so, as the magnitude of the response to stimulation can vary between subjects and implanted hardware. This is highlighted in **Figure 2.5B**, where damage to the recording electrode decreased the overall amplitude of the signal. So, while the estimated optimal stimulation parameters are the same, the overall magnitude of the response differs.

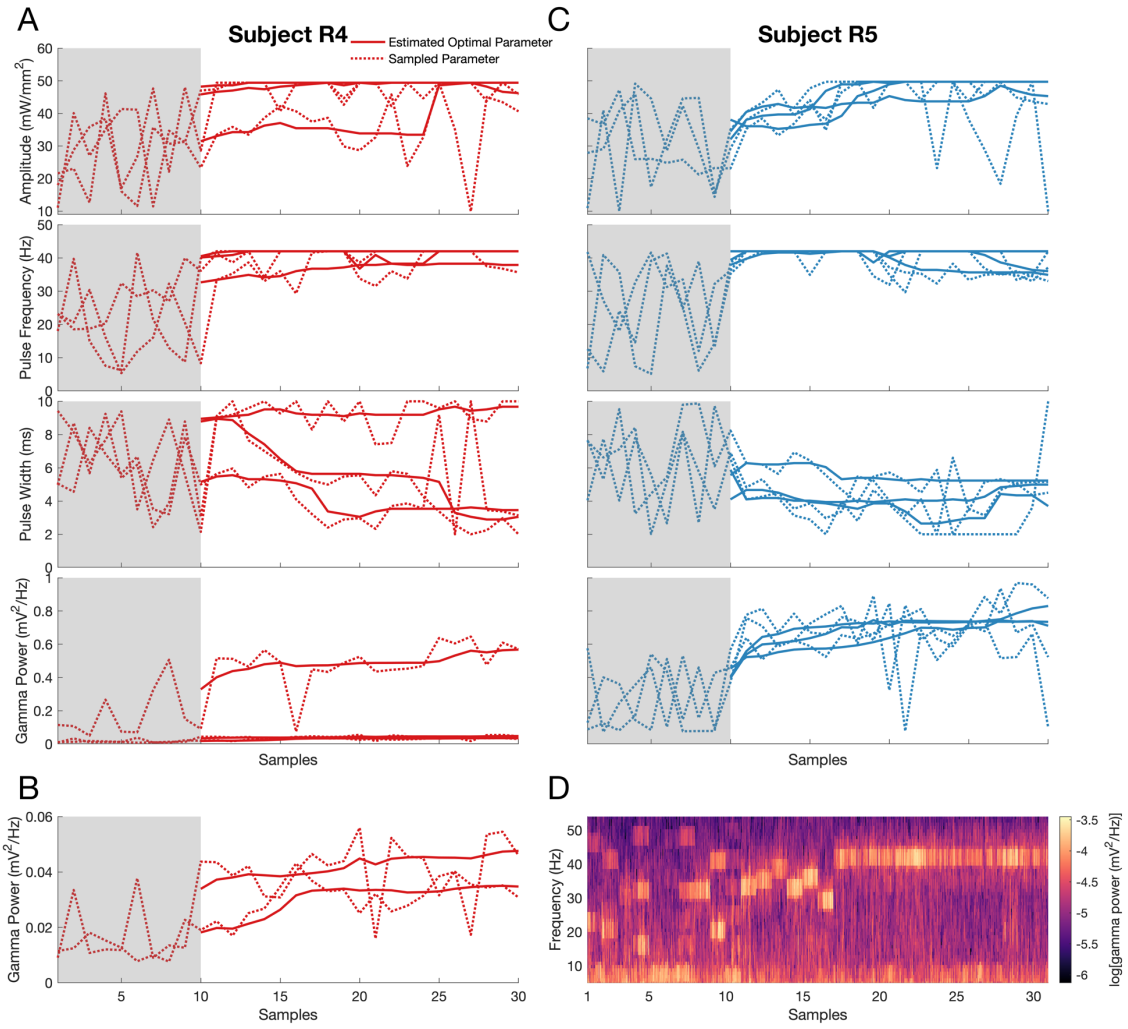

Figure 2.5: Trajectory of maximization of hippocampal gamma power *in vivo*.

(A) Stimulation parameter search trajectory and the effect of stimulation on hippocampal gamma power for three trials in subject R4. Solid lines indicate the optimization algorithm's estimate of the optimal stimulation parameters and their predicted effect, while the dotted line shows the actual stimulation parameters sampled and the actual hippocampal gamma power induced. Grey box indicates the 10 samples that were collected as part of the burn-in period for BaO. (B) Re-scaled effect of stimulation on hippocampal gamma power to show two trials with reduced signal amplitude for subject R4. (C) Corresponding stimulation parameter search trajectory and effect on hippocampal gamma power for subject R5. (D) Spectrogram of the hippocampal LFP from a representative optimization trial in subject R5. One second non-overlapping window, with post stimulation washout periods omitted.

2.3.3.2 Validation phase

After the optimization phase and a short delay to allow for the washout of any potential accumulated effects, each subject underwent a post-optimization validation-phase. Overall, the estimated optimal parameters produced a hippocampal gamma power 2-2.5 times that during sham stimulation (**Figure 2.6A**). The stimulation parameters estimated to have the worst performance on the objective function (either no effect or a decrease), did not induce a statistically significant increase compared to sham stimulation (two-sample t-test, p-value > 0.05). **Figure 2.6C** shows the mean wideband power spectral density (PSD) for the three conditions from a representative trial. For the estimated optimal stimulation parameters there is a large peak at 36.7 Hz corresponding to the applied stimulation frequency, along with its harmonic at 73.1 Hz. For the estimated worst stimulation parameters, there are harmonics at approximately every 15 Hz. As described in our previous work, these harmonics are expected at multiples of the stimulation parameters [40]. No such peaks or harmonics are seen for the PSD under sham stimulation. Focusing on the gamma range in **Figure 2.6D** there is a large peak for the estimated optimal stimulation pattern and a much smaller peak for the estimated work stimulation pattern.

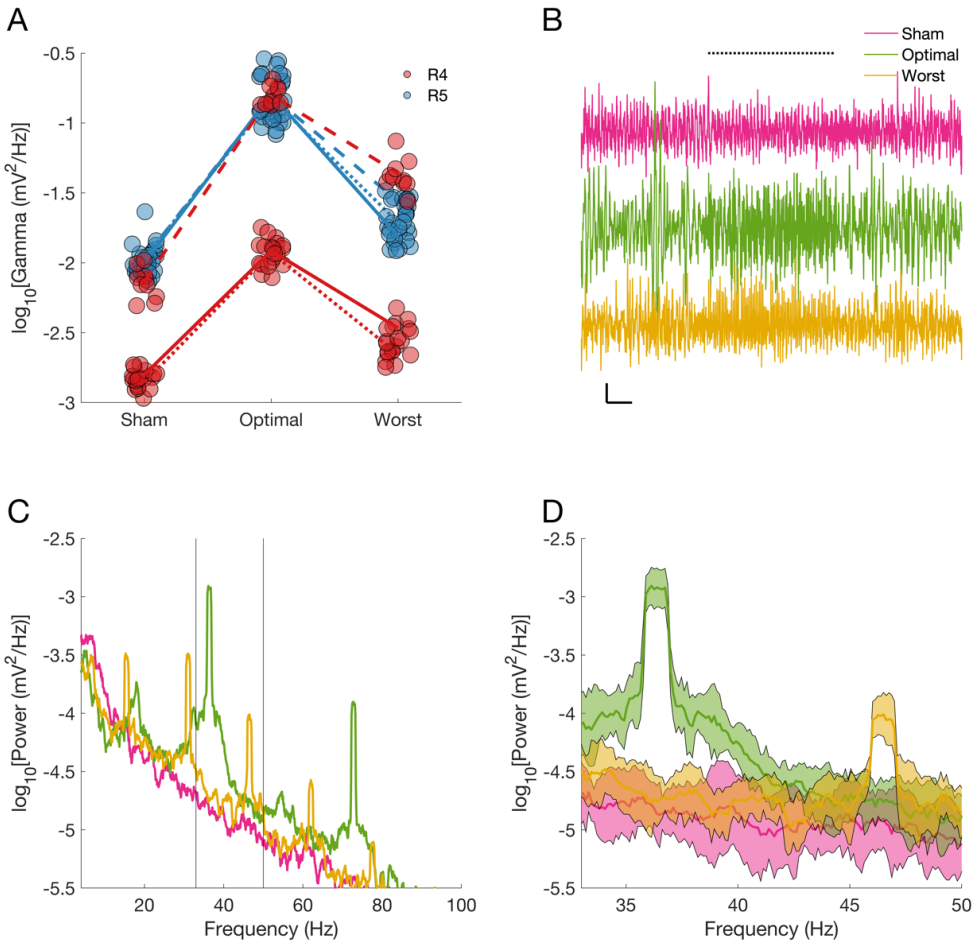

Figure 2.6: Validation phase for stimulation parameters to maximize hippocampal gamma power.

(A) Gamma power for each randomized trial of the three conditions: sham, estimated optimal, and estimated worst stimulation pattern during the validation phase. Lines connect the mean hippocampal gamma power across the 10 samples for each trial/subject. (B) LFP from under the three conditions from a single channel during a representative sample. Dotted black line shows the period of time when stimulation was applied, scale bars for 1s and 0.1mV. (C) Average power spectrum for each of the three conditions from a representative testing phase. (D) Power spectrum focused on the gamma frequency range. Shaded regions indicate the standard deviation.

2.4 Discussion

2.4.1 Optimization framework recap

Selecting neural modulation parameters is a challenge that spans much of neuroscience, neurology, neurosurgery, even psychiatry. Evaluating the effects of any neurological intervention is time and resource intensive, all the more so with neural modulation therapies where the high dimensionality of stimulation parameters makes the problem of finding the optimal choice even more difficult, if not intractable. The optimization framework presented here offers a principled approach to address this problem. Starting with exploratory data, users can construct ground-truth models that allow for optimization algorithms to be rapidly and repeatedly prototyped and evaluated using a simulated system with the necessary black-box characteristics. As we have shown, these algorithms can be directly applied in real-time to learn subject-specific optimal stimulation parameters.

2.4.2 Developing a platform for prototyping data-driven optimization algorithms

The first step in implementing this framework is the construction of the ground-truth models. While any optimization algorithm can be deployed *in vivo* without being evaluated on ground-truth models, this greatly increases the risk of an unsuccessful optimization and provides little to no recourse for interpreting the failure. Here, our data was already collected using a pre-specified grid. However, a random search or importance based sampling could be potentially used to greater effect. Moreover, the structure used for modeling the data is not limited to a Gaussian process. Alternative model structures that can be used include random forests, [46] recently developed deep neural network

architectures that directly model noise and uncertainty, [47] and biophysical models. Ultimately, selecting the model structure and fitting approach should be guided by the optimization problem in question.

2.4.3 Designing an optimization system for neural modulation

By using the simulation models for prototyping, the framework predicted that among those tested, the UCB(0.4) algorithm would perform best at maximizing hippocampal gamma power. While the predicted performance of UCB(0.4) was born out the *in vivo* optimization and validation experiments, this does not fully highlight the predictive value of the framework. When prototyping algorithms, a version of the UCB acquisition function with decreased exploration, UCB(0.01), performed worse than UCB(0.4) as it would prematurely converge to a suboptimal set of stimulation parameters. This behavior was correctly predicted by the framework in a small set of control experiments (Chapter 2, Section 5.4), demonstrating that it could detect the nuanced differences between these two algorithms.

There are several factors that can prevent successful optimization. In particular, optimization is only possible if the selected parameter space can have the desired effect on the objective function. For example, all stimulation settings within this parameter space caused an increase in hippocampal gamma power, such that it would be futile to search for stimulation settings to decrease hippocampal gamma. Another critical factor is the correct engagement with the neural tissue and accurate measurement of the objective. One example was shown in the optimization to maximize hippocampal gamma power in subject R5.

After the first trial, the recording electrode was damaged which resulted in an overall lower signal amplitude, decreasing hippocampal gamma under baseline and stimulation conditions. Had the electrode suffered further damage or been incorrectly positioned during surgery, there would be no signal to optimize. The potential for changes in the underlying biophysical system (electrodes, behavioral state, etc.) to influence the effect of stimulation parameters on the objective function highlights the importance of our ongoing work to develop optimization approaches that can adapt to changing situations [48]. In sum, while data-driven optimization can be a powerful tool, it is only as effective as the combination of the parameter space, the objective function, and engagement with the neural tissue.

An important decision in the development of this framework was the experimental model. The medial septum optogenetic stimulation model offered several key advantages. First, the relationship between the pulsatile stimulation parameters and hippocampal activity was characterized in our previous work. In [39], we showed that pulsatile stimulation at a given frequency induces evoked potentials in hippocampus at that frequency. In the frequency domain, this creates a sharp peak at the stimulation frequency and harmonics at multiples of that frequency. Then, as shown in [38], we can reasonably anticipate the stimulation parameters that will maximize a spectral power feature (i.e. gamma). Combined with the consistency between subjects, this allowed us to independently verify the results of the search algorithm when applied to an unknown subject. The lack of subject-to-subject variability does differ from the clinical scenario where each patient has a different optimal stimulation setting. However, through the setpoint experiments and benchmarks in Chapter 2, Section 5.5 we were able to show that

the designed algorithm, UCB(0.4), could generalize across multiple different objective functions.

While the experimental model was well-suited to the development of this framework, it also limits some of the conclusions that can be drawn from this study. Based on the findings in [38] we chose to use gamma power as the neural feature to maximize. However, while medial septum optogenetic stimulation acts via biological conduction along the septohippocampal circuit, it is not necessarily physiological. In other words, the increase in gamma power from the optimal stimulation is not equivalent to naturally elevated gamma during certain tasks [49]. With these limitations in mind, this experimental model can serve as a valuable toy problem for prototyping novel neural modulation algorithms, just as the MNIST dataset is for machine vision [50]. Another limitation of this study was the parameter space used. While we restricted ourselves to a fairly narrow frequency band to ensure proper functioning of our opsin [51], future work will explore how data-driven optimization behaves with larger and more complex parameter spaces and objective functions in this model. Finally, the specific stimulation protocol differed between the initial data collection experiments (20 seconds of stimulation followed by 60 seconds of washout) and the real-time optimization experiments (5 seconds of stimulation followed by 25 seconds of washout). While a potential disconnect, the UCB(0.4) algorithm designed based on the longer stimulation protocol performed similarly and identified equivalent optimal stimulation settings in the real-time experiments using the shorter stimulation protocol. However, more substantial changes to the experimental protocol

could warrant a re-design of the optimization algorithm by repeating the steps of the framework.

The purpose of this demonstration is not to provide a solution, e.g., UCB(0.4), to a narrowly defined problem – modulating hippocampal gamma power with optogenetic stimulation of the medial septum – but to instead provide a roadmap to guide solutions for a broader problem that permeates all of neural modulation. While the implementation of the framework presented here centered around optogenetic stimulation, which does not have an immediate pathway to clinical translation, there is no reason this framework cannot be extended to clinical therapies. Several extensions of the standard optimization algorithms may find specific application in the clinical domain. These include: multi-objective optimization where multiple objectives are mutually competitive and need to be addressed simultaneously [52]; safe optimization where regions of the parameter space may be unsafe but the location of these regions is unknown a priori [53]; and preferential optimization where the objective function is more subjective [54]. Finally, there is state-dependent optimization, where the algorithm learns a stimulation policy that depends on the current state of the neural system. Taken to its logical conclusion and combined with approaches from control theory, state-dependent optimization is a direct pathway to the application of optimal control techniques to neural modulation. Ultimately, using this framework as a starting point to guide the use of exploratory data and simulation models, data-driven optimization approaches are well-poised to improve the process of tailoring neural modulation therapy to specific patient- and therapy-dependent requirements.

In the next several chapters, I will show how this framework can be extended to different applications of neural modulation, each with unique properties and constraints.

2.5 Supplement

2.5.1 Pulse width has a slight effect on induced hippocampal gamma power

Figure 2.7 shows the effect of pulse width on gamma power for subject R1. Three independent 2 dimensional models were constructed using the stimulation parameters and measured gamma power for each of the pulse widths (2, 5, 10ms) tested in the grid search experiments. This pattern was observed in the other subjects (R1 and R2, data not shown) as well.

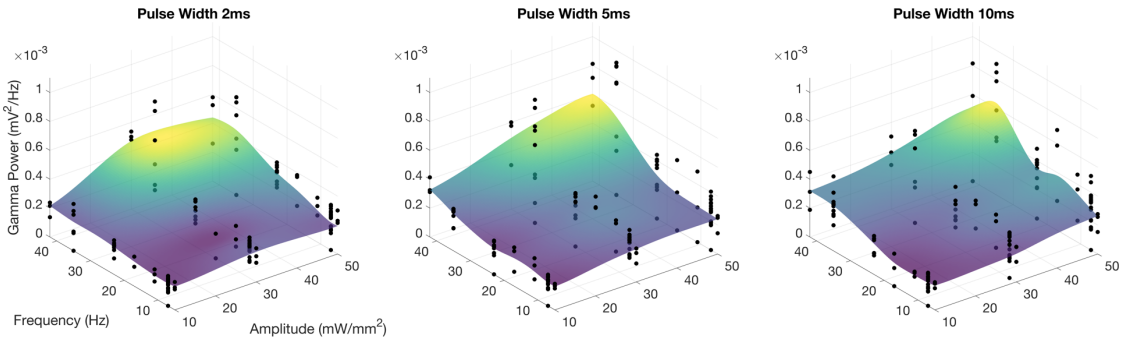

Figure 2.7: Modeled data at different pulse width cross-sections.
Surfaces show the expected gamma power a model fit to the data for specific pulse widths. Each dot indicates the stimulation parameters and measured gamma power. The color scale is for visualizing contrast within models.

2.5.2 Control animals without ChR2 opsin demonstrate lack of photoelectric effect

A major consideration in any neural modulation experiment is differentiating stimulation artifacts from real electrophysiological signals. This is especially important

when optimizing stimulation based on an electrophysiological objective function. If the optimization algorithm is detecting the artifact instead of the real signal, it risks optimizing the artifact instead of the desired electrophysiological signal.

To control for this, two animals were transfected with a virus containing just the hSyn promotor and the eYFP fluroescent indicator. Subjects underwent the same surgical procedure and grid search experiments as described in the methods. Similarly, for each stimulation the gamma power was measured and used to create a simulation model.

Figure 2.8 shows the expectation estimated from the models created for the two control subjects, and subject R1. By plotting the two models on the same scale, it can be readily observed, from the perspective of the gamma simulation model, there is no response to stimulation in the control subjects. This is evident in the different amplitudes of the models, and the inconsistent shape of the control models compared to the three models with the opsin (**Figure 2.2**). This supports the lack of stimulation artifact in the objective function measured from the hippocampus.

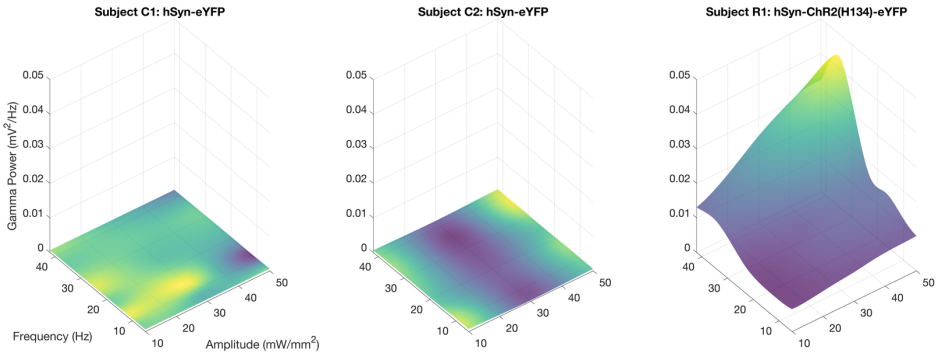

Figure 2.8: Ground truth models for control subjects.

Each surface is the estimated expectation interpolated across the amplitude-frequency input space. The color scale is for visualizing contrast within models.

2.5.3 Simulation results for subjects R2 and R3

Simulation experiments were completed for each of the models created using data from each of the three subjects. **Figure 2.9** shows the final error of the simulation results from subjects R2 and R3.

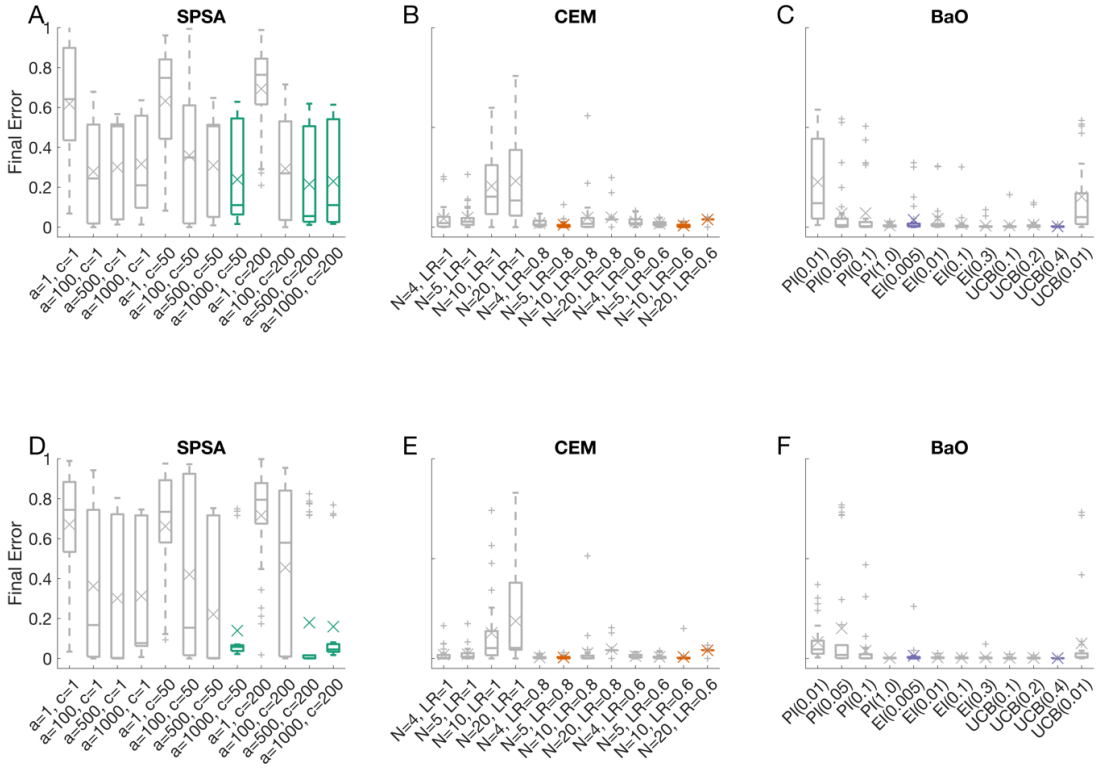

Figure 2.9: The final error statistics for running the optimization algorithms under different configurations on subject models R2 and R3.

Boxplots show the quartiles and the 'x' indicates the mean error. The labels on the x-axis ticks describe the specific configuration used for each optimization experiment. The highlighted results indicate that for a given optimization class, that specific configuration had the lowest final error for at least one of the subjects.

2.5.4 Framework predicts optimization performance for different algorithms/configurations

In the main body of this chapter, we demonstrated using this framework to design *in silico* and deploy *in vivo* an optimization algorithm for finding the medial septum optogenetic stimulation parameters that maximize hippocampal gamma power. To reinforce this, we further validate that the results of the *in silico* experiments could accurately predict the performance of a given optimization algorithm when deployed *in vivo*.

The BaO UCB acquisition function has a single configuration parameter η, which controls the relative exploration of the algorithm. When η is high, the algorithm will weigh uncertainty more heavily than the expected output of the objective function. When η is low, the acquisition function will preferentially select stimulation parameters expected to produce high gamma power with little uncertainty. As a consequence, the algorithm will not explore the parameter space, but will instead prematurely converge to the best stimulation parameter observed during the burn-in phase. Therefore, we would expect the *in silico* experiments to show faster convergence and higher overall error with the UCB(0.01) algorithm than with UCB(0.4), which should then be recapitulated when deployed *in vivo*.

In this supplementary experiment (2.5.4), we directly compare the performance of UCB(0.01) and UCB(0.4) *in silico* and *in vivo*. The simulation experiments were conducted as described in the methods with the exception that each optimization trial only used 30

samples, as opposed to 100 samples. The data model for subject R1 was used. The *in vivo* optimization and validation phases were performed using subjects R4 and R5 as described in the methods for an additional trial using the UCB(0.01) algorithm.

Figure 2.10 compares the *in silico* results of using the UCB(0.01) algorithm compared to UCB(0.4). **Figure 2.10A** shows the error trajectory averaged by sample for the two algorithms. While both start at the same mean error of 0.4, the error for the UCB(0.4) algorithm decreases much more rapidly and persistently. In contrast, the UCB(0.01) algorithm error levels off faster and after 30 samples has a much higher final error. **Figure 2.10B** compares the overall distribution of the final error for the two algorithms. It can be seen that, not only does UCB(0.4) have an overall lower error, the variance of the distribution is also smaller. In contrast, the range of final errors for UCB(0.01) extends all the way to ~0.6, while the worst outlier for UCB(0.4) is just above 0.3. Overall, these findings show that, in simulation, the framework predicts that UCB(0.01) will perform worse than UCB(0.4).

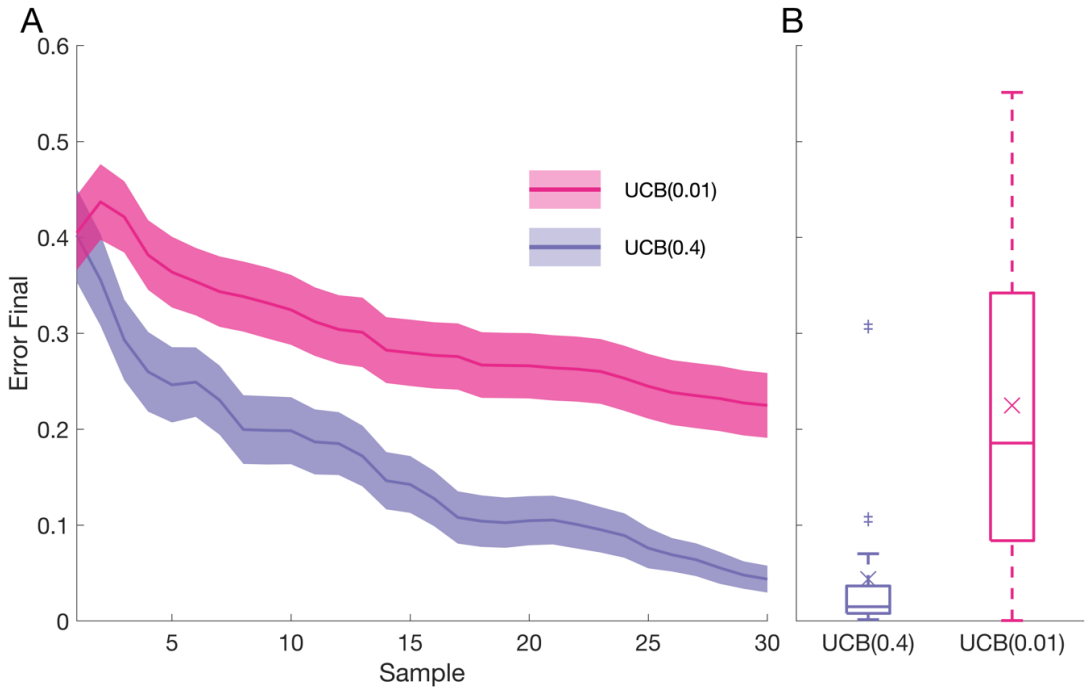

Figure 2.10: In silico performance of UCB(0.4) vs. UCB(0.1).

(A) Per sample error of the optimal stimulation parameter estimated by the algorithms. Center line and shaded regions indicates the per sample mean and standard error, respectively, across 30 trials. (B) Boxplot showing the range and quartiles of the final error. Outliers marked with +, and X indicates mean including outliers.

Figure 2.11A shows the *in vivo* optimization phase for subject R4. In the amplitude trajectory, we can clearly see that UCB(0.01) has rapidly converged to a lower amplitude, 20 mW/mm^2, compared to the assumed optimum of 45-50 mW/mm^2. The lack of exploration by UCB(0.01) is also evidenced by the consistency in the sampled amplitude compared with the estimated optimal. While UCB(0.01) found the expected optimal frequency, and pulse width does not have a profound effect on the induced gamma power, the premature convergence to the lower amplitude resulted in a significantly lower induced

gamma power. The poor performance of the stimulation parameters estimated by UCB(0.01) were verified during the validation phase (**Figure 2.11B**).

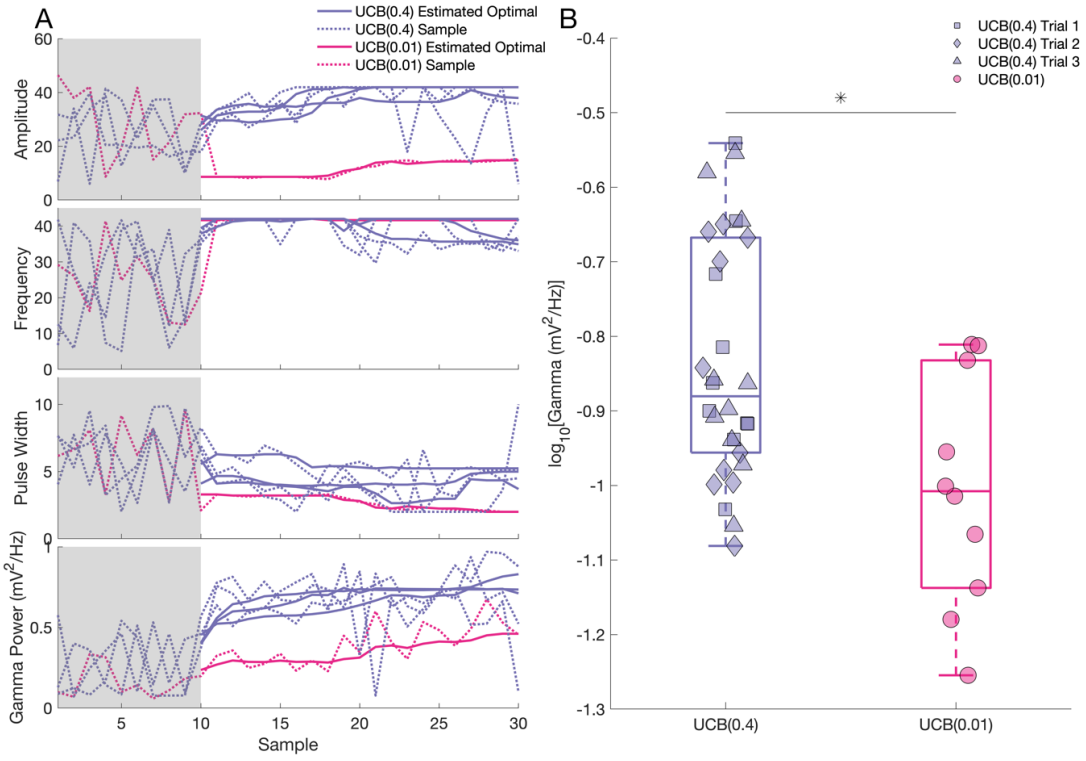

Figure 2.11: Performance of UCB(0.01) compared to UCB(0.4) in vivo.

(A) Stimulation parameter trajectories of the two algorithms during the optimization phase. The solid lines indicate the per sample optimal value of the stimulation parameters estimated by the algorithm, while the dotted lines indicate the actual stimulation parameters sampled. The purple lines show the optimization results using the UCB(0.4) algorithm as previously shown in **Figure 2.4**, while the pink lines are the results using UCB(0.01). The gray box indicates the 10 burn-in samples at the beginning of each optimization trial. (B) The induced hippocampal gamma using the estimated optimal stimulation parameters. Box plots show the range and quartiles for the three trials using UCB(0.4) (purple) and UCB(0.01) (pink). The marker shapes indicate the validation results for each of the 4 trials shown.

2.5.5 Performance of UCB(0.4) on subject data and benchmark models

While the main focus of the framework is on designing optimization algorithms for specific neural modulation problems, it is also important to characterize the limitations of the resulting optimization algorithm. To demonstrate this, we evaluated the performance of the UCB(0.4) optimization algorithm on three representative benchmark objectives along with the model derived from subject R1.

The three objectives used were, a quadratic/sphere objective:

$$f(x) = -1 * \sum_{d=1}^{D} x_d^2 \tag{2.16}$$

The Rastrigin objective:

$$f(x) = \sum_{d=1}^{D} 10 + x_d^2 - 10 * \cos(2\pi * x_d) \tag{2.17}$$

And the Easom objective:

$$f(x) = \left(\prod_{d=1}^{D} \cos(x_d * 30) \right) e^{\sum_{d=1}^{D} (3\pi - x_d * 30)^2} \tag{2.18}$$

For each objective, we used 1, 2, and 3-dimensional (D) variations and compared with model for subject R1 containing the frequency (1D), the frequency and amplitude (2D),

and the frequency, amplitude, and pulse width (3D). For each objective/model, we adjusted the measurement noise to sweep nine different values of the contrast-to-noise ratio (CNR) [0.1, 0.5, 1, 1.5, 2, 3, 4, 5]. The UCB(0.4) algorithm was evaluated on each model/CNR combination for 30 trials of 100 samples each, as described in the methods.

Figure 2.12A shows the three 1D benchmark objectives and model for R1. From the 1D variations of these objectives, the difference in the difficulty can be seen. The simplest, sphere/quadratic function, has a single maximum and is overall relatively simple. Slightly more difficult is the model from subject R1, which has two local extrema, one of which is much higher than the other. The third objective, the Rastrigin function, has many local extrema over its domain, of which many are similar, but only one is the global maximum. Finally, the Easom function is relatively flat across much of its domain, with a sharp spike to create the global maximum. In **Figure 2.12B**, the final error and convergence rate for UCB(0.4) is shown for each of these functions as the CNR is adjusted. Across CNR values, the sphere/quadratic objective and model for subject R1 perform similarly, although at lower CNR levels, they have relatively high errors, but still show some optimization after 100 samples. The UCB(0.4) performs slightly worse with the Rastrigin function at lower CNR levels, but more comparably to the sphere and R1-frequency models as the CNR increases. For the Easom model, the algorithm performs poorly and variably at low CNRs. The performance of the algorithm increases with CNR, but still worse than with the other three.

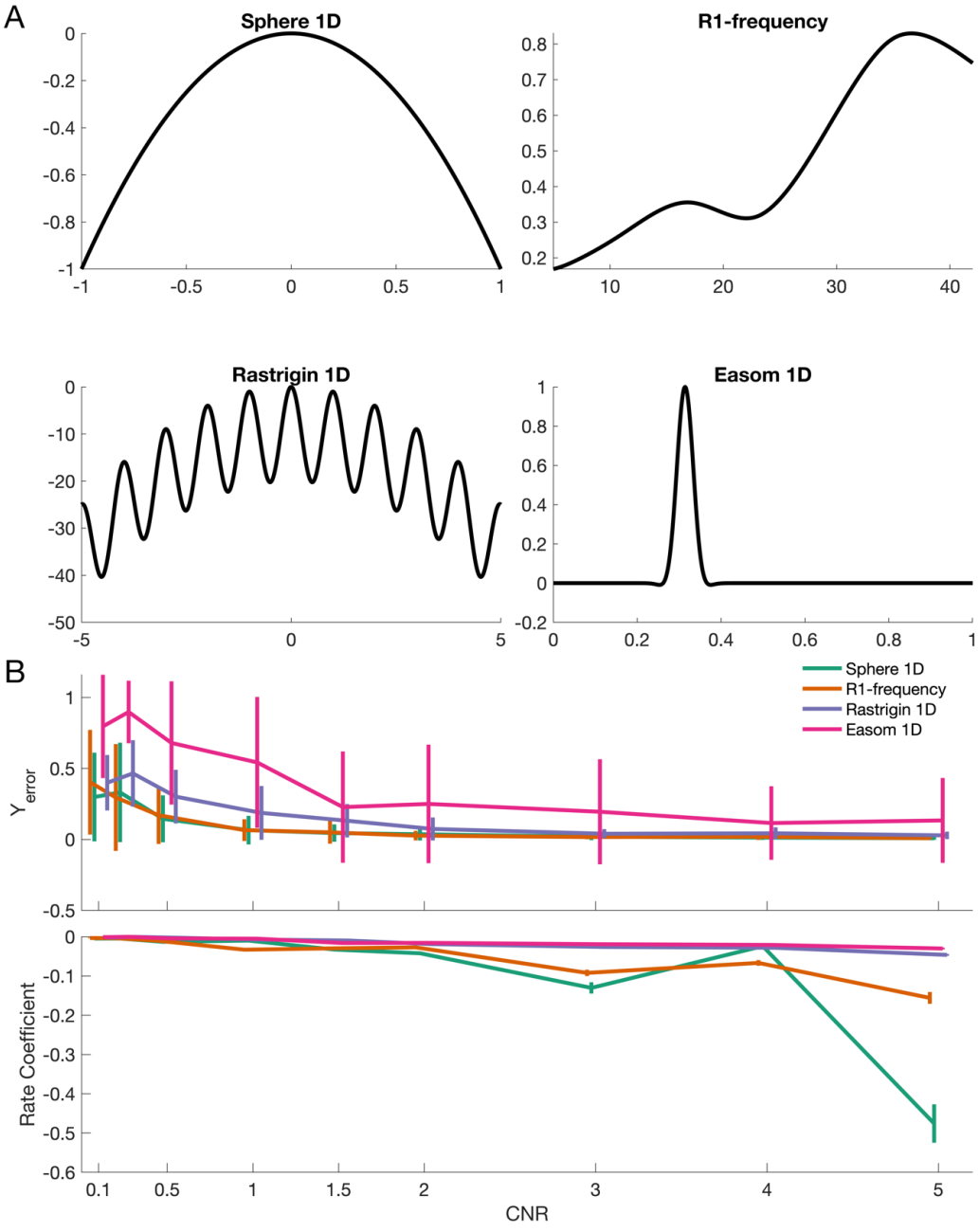

Figure 2.12: Performance of UCB(0.4) on 1 dimensional benchmark functions.

(A) 1D variations of the 3 benchmark objective and the frequency model for subject R1. (B) The final error and convergence rate of the UCB(0.4) algorithm as the CNR is increased. Vertical bars indicate the standard error across 30 trials.

Figure 2.13A shows the 2-dimensional counterparts for each of the three benchmark objectives and the frequency/amplitude model for subject R1. While mostly similar to the 1D versions, the Easom model becomes even more difficult as the main region of variability covers a much smaller proportion of the parameter space. This increased challenge is reflected in **Figure 2.13B**, where the UCB(0.4) algorithm completely fails to find the optimum for the Easom 2D objective at low CNR values, and only occasionally finds the optimum at higher CNR levels. The other two benchmark objectives and the subject model perform similarly as in the 1D cases, albeit with more stratification in the performance at lower CNRs where the algorithm performs best on the sphere/quadratic object, next best on the model for subject R1 and third on the Rastrigin model.

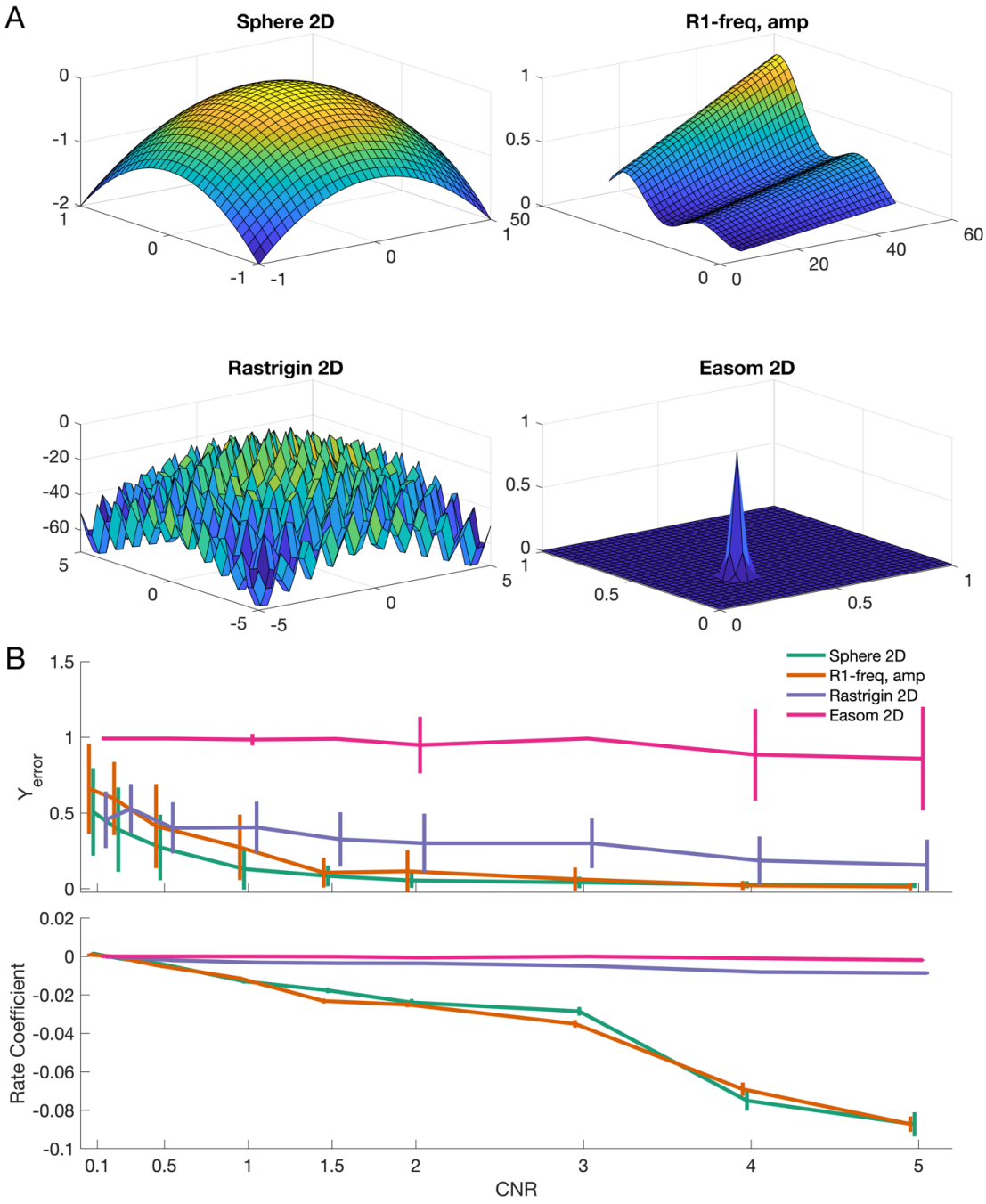

Figure 2.13: Performance of UCB(0.4) on 2 dimensional benchmark functions.

(A) 2D variations of the 3 benchmark objective and the frequency model for subject R1. (B) The final error and convergence coefficient of the UCB(0.4) algorithm as the CNR is increased. Vertical bars indicate the standard error across 30 trials.

While it is difficult to visualize the 3D version of each of the benchmark objectives, **Figure 2.14A** shows that the overall trend is preserved. The Easom models becomes completely unoptimizable within 100 samples at any CNR, while the others become more tractable with increasing CNR. As in the 2D case, the algorithm performs comparable on the sphere objective and model for subject R1, and slightly worse for the Rastrigin model.

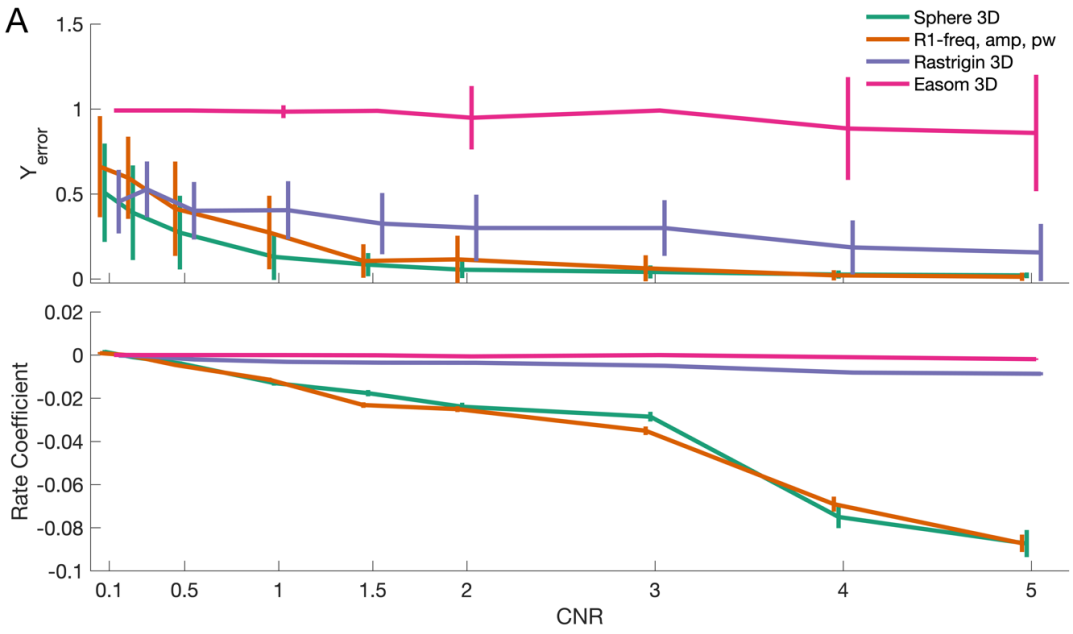

Figure 2.14: Performance of UCB(0.4) on 3 dimensional benchmark functions.

(A) The final error and convergence rate of the UCB(0.4) algorithm as the CNR is increased. Vertical bars indicate the standard error across 30 trials.

Ultimately this experiment highlights the different factors that make an objective function challenging for an optimization algorithm. First, as CNR increases, the responses from the objective become less noisy and it becomes easier to find the global maximum of the objective. Additionally, the smoothness of the objective plays a significant role. Both the Rastrigin and the Easom models have very sharp changes across the parameter space, and as such prove more challenging for the UCB(0.4) algorithm. This is especially true for the Easom model, as so much of the parameter space is flat, and give little information to the algorithm. Without accidentally sampling close to the global maximum, it is unlikely it will find the optimal parameters. Lastly, all objectives became more difficult when the dimension was increased from 2D to 3D, a trend that will undoubtably continue as higher dimensional objectives are designed.

2.5.6 Viral expression and electrode placement

After the completion of the study, histology was performed to verify the location of the electrode and viral vector expression. Rats were transcardially perfused with 0.9% saline followed by 4% paraformaldehyde in 0.1M phosphate buffer after they were deeply anesthetized with a lethal dose of Euthasol (100mg/kg, Virbac, Fort Worth, Tx, USA). Rat heads were post-fixed overnight with the electrodes and ferrules in place. Brains were subsequently dissected out and transferred to 30% sucrose in phosphate buffered saline until they sank. Then, brains were sectioned coronally at 40 μm using a cryostat and mounted on glass slides with Vectashield DAPI (Burlingame, CA, USA) mounting medium for visualization of nuclei. Slides were imaged to visualize the fluorescent tags in the opsins with a Leica DRME fluorescent microscope equipped with a CCD camera and

SimplePCI software (Leica Microsystem, Buffalo Grove, IL). Hippocampal sections were mounted and stained using Cresyl Violet (C5042 Sigma-Aldrich, St. Louis, MO), coverslipped and imaged using a Nikon DS-fil color digital camera on a Nikon E400 microscope and NIS-Elements software (Nikon Instruments, Inc., Melvile, NY, USA).

Expression of the GFP opsin reporter was visible in neurons in the medial septum (**Figure 2.15A**) as well as in processes and terminals in many areas of the hippocampus (**Figure 2.15B**) including the stratum oriens, stratum radiatum (**Figure 2.15C**) and the hillus of the dentate gyrus (**Figure 2.15D**). The electrode track was visible in the CA1 layer of the hippocampus (**Figure 2.15E**).

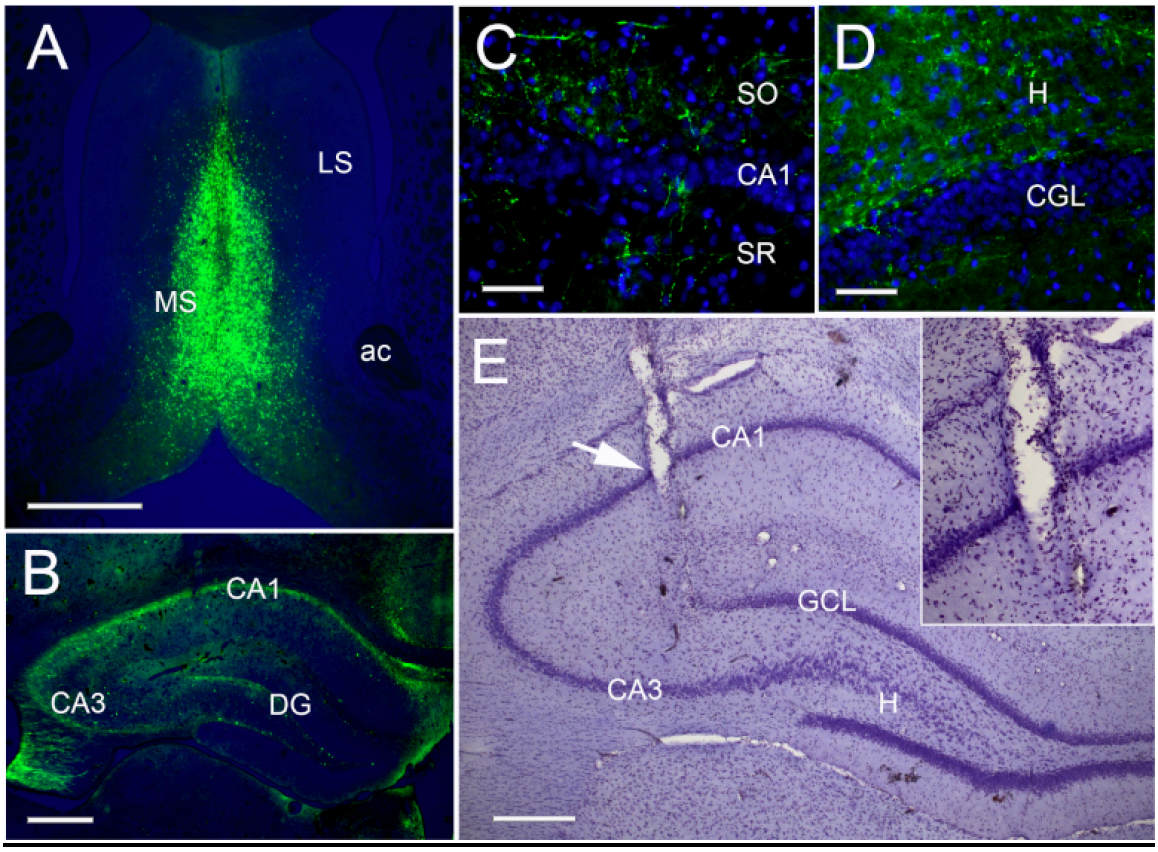

Figure 2.15: Viral expression and electrode placement.

(A) Micrograph through the septal area showing GFP opsin reporter expression in neurons in the medial septum. (B) Micrograph through the dorsal hippocampus showing GFP opsin reporter expression in many areas of the hippocampus. (C) Micrograph in the CA1 region showing GFP opsin reporter expression in processes and terminals in the stratum oriens and stratum radiatum. (D) Micrograph showing GFP opsin reporter expression in processes and terminals in hillus region of the DG. (E) Micrograph of Cresyl Violet stained section through the hippocampus showing an electrode track (arrow) in the CA1 pyramidal layer. Abbreviations: MS: medial septum, LS: lateral septum, ac: anterior commissure, CA1: pyramidal cell layer 1, CA3: pyramidal cell layer 3, DG: dentate gyrus, SO: stratum oriens, SR: stratum radiatum, H: hillus, GCL: granule cell layer. Scale: A, B and E: 1 mm; C and D: 80 μm.

CHAPTER 3. APPLICATION OF THE FRAMEWORK FOR SETPOINT OBJECTIVES

3.1 Introduction

In Chapter 2 we designed and characterized and optimization algorithm to find the medial septum optogenetic stimulation settings that maximize hippocampal gamma power. However, it may not be ideal, or even safe, to maximize a specific neural feature or behavioral measures. In fact, the goal may be to find the stimulation setting that modulates the electrophysiological or behavioral measurement to a specific value somewhere between the maximum and the minimum.

In the second application of this framework, we reframe the task to finding the stimulation parameters that modulate hippocampal gamma to a desired setpoint. We first adapt our framework for setpoint objectives, and use the models constructed in Chapter 2 to further explore the implications of this change. We then evaluate the UCB(0.4) algorithm on this setpoint optimization problem, before validating it *in vivo*.

3.2 Methods

3.2.1 Surgery, virus injection, and electrode implant

Surgical details are identical to those described in Chapter 2, Section 2.1

3.2.2 Setpoint objective function

According to the first step in this framework, we return to the ground-truth models mapping stimulation amplitude (a), frequency (f), and pulse-width (p), to induced hippocampal gamma power (Y_{gamma}):

$$Y_{gamma} = F(a, f, p) \tag{3.16}$$

To construct a setpoint objective function we apply the following transformation,

$$-1 * abs\left(Y_{gamma} - Y_{SP}\right) \tag{3.17}$$

where Y_{SP} is the desired setpoint. This transformation has the effect of centering the model so that the setpoint is now at 0, and "folding" the model so that regions where the output is above the setpoint are flipped about the zero-axis.

3.2.3 Simulation experiments

The BaO algorithm found to have the best performance for the maximization experiments in Chapter 2 was used for the setpoint optimization experiments. This algorithm was used to find the max of the setpoint objective applied to the simulation model for subject R1. For each of three setpoints (0.15, 0.45, and 0.8), 30 trials were performed with 100 samples each as in the simulation experiments for gamma maximization.

3.2.4 In vivo experiments

Setpoint optimization was validated in two subjects (R5 and R6). Here, the objective function was set to find the stimulation parameters that modulate hippocampal gamma power to either 1.5, 3.0, or 20 times a baseline measurement. The subjects were connected to the data acquisition system and LED stimulator as described for the gamma maximization experiments. The LFP was recorded for 5 minutes at baseline and hippocampal gamma power was calculated on non-overlapping 5 second segments. The median value was then used as the baseline for the rest of the experiment.

The optimization phase consisted of 10 burn-in samples followed by 30 samples using UCB (0.4). Each sample consisted of 5 seconds of stimulation followed by a 25 second washout. For the validation phase, the estimated optimal stimulation parameters for the three setpoints were applied in a random order ten times each for a total of 30 samples. Sample timing was as in the optimization phase.

3.3 Results

3.3.1 Setpoint objective applied to ground-truth models

The application of the setpoint transformation to the model mapping stimulation parameters to hippocampal gamma power is shown in **Figure 3.1**. **Figures 3.1A-C**, show the setpoint objective functions at three different setpoints (0.15, 0.45, 0.8, where 1.0 is the maximum gamma power) for subject R1. It can be seen that, rather than a single stimulation parameter configuration that maximizes the setpoint objective function, there is a level set

of stimulation parameters that would modulate hippocampal gamma to the desired setpoint. When the original Y_{gamma} model is projected onto a 2D surface, this level set can be drawn as a contour line where the model outputs, Y_{gamma}, are equal. Based on this, we would expect that optimization using this objective function could converge to a range of different, but effectively equivalent, stimulation parameters. When comparing the objectives for the three different setpoints it can be seen that the width of the band around the contour lines varies. When the setpoint is 0.15, the band is very wide showing that there are a number of stimulation patterns that have an effect that is near that of those on the level set of optimal solutions. In contrast, for the objective function with a setpoint of 0.45, the width of the band is narrower. **Figure 3.1D** compares the level set solutions for the three setpoints across the three subjects R1-3. For a given setpoint, the level set solutions for the three subjects are in the same general region of the parameter space. However, there is a degree of subject-specific-variability. This indicates the need for an optimization approach to learn subject-specific stimulation parameters.

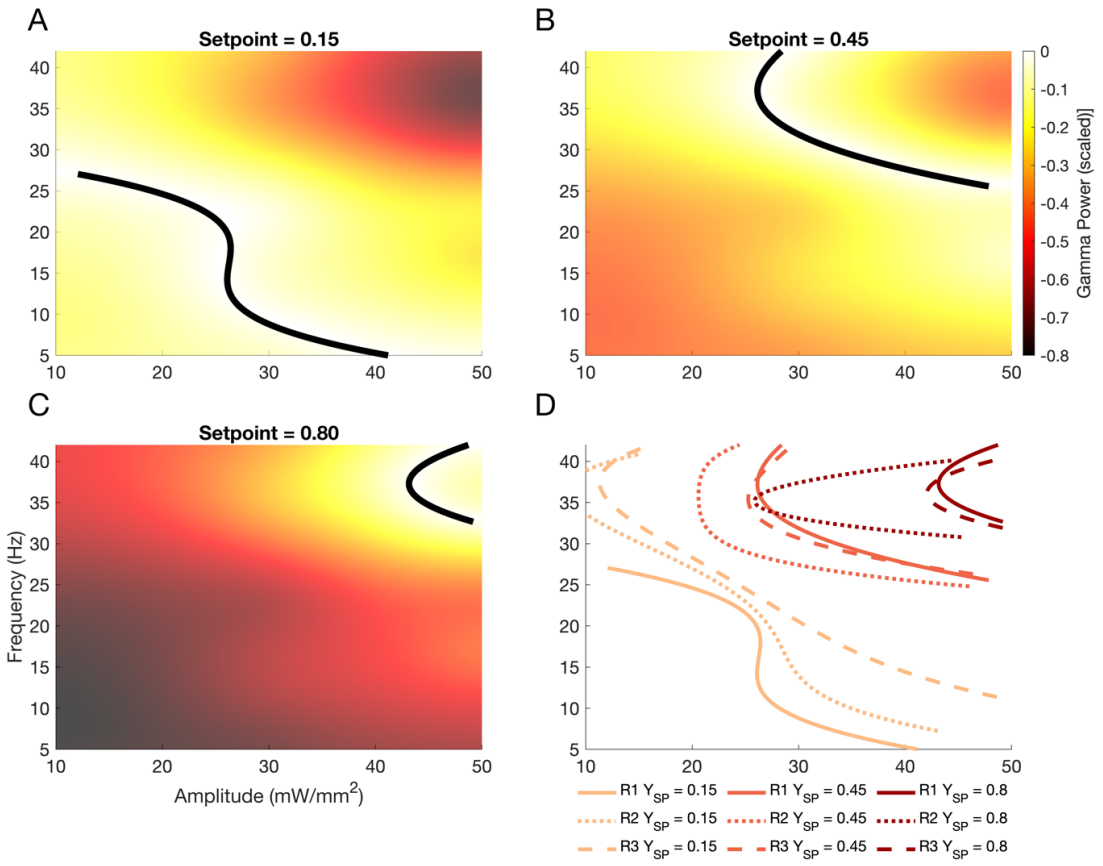

Figure 3.1: Setpoint objective functions can have a level set of equivalent solutions.
(A-C) The three setpoint objective functions for subject R1. Black line is a polynomial regression along the top 100 objective values and is intended for illustration, not an exact representation of the level set solution. (D) The three set point objective functions for R1 (solid lines), R2 (dashed lines), R3 (dotted lines).

3.3.2 Using framework to prototype optimization for a setpoint objective

We then adapted the previously described simulation platform to prototype data-driven optimization for a setpoint objective function. Building on our previous results, we focused our analysis on the UCB(0.4) algorithm. **Figure 3.2A** shows the search trajectories for each setpoint using the model constructed from subject R1. Looking at a single representative trial (bold), for each of the setpoints, we can see that the optimization

algorithm converged on stimulation amplitudes and frequencies along the corresponding level set. Over the course of the optimization, there are samples in which the estimated optimal frequency and amplitude change somewhat dramatically as they switch to a nearly equivalent point on the level set. However, given that the simulation platform can rapidly run multiple trials under identical conditions, the overall range of behavior of the optimization system could be observed. As expected, across 12 independent trials for a setpoint the optimization algorithm converged to a range of different stimulation parameters, all of which had a similar effect on hippocampal gamma power. The optimal parameters estimated by the optimization system compared to the ground truth level set solutions are shown in **Figure 3.2B**. For the highest setpoint, the optimal estimated parameters are grouped tightly around the level set solution, while the lower two setpoints have a wider distribution. This is likely due to the flatness of the objective function as described in the analysis of the ground truth model. **Figure 3.2C** shows the effect of the stimulation parameters selected by the optimization system when applied to the ground truth model. For both the low (0.15) and high (0.80) setpoints, the optimization algorithm was able to reliably identify stimulation parameters that modulated hippocampal gamma to the desired value. However, for the middle setpoint (0.45), there was more variability in the effect on hippocampal gamma power from trial-to-trial.

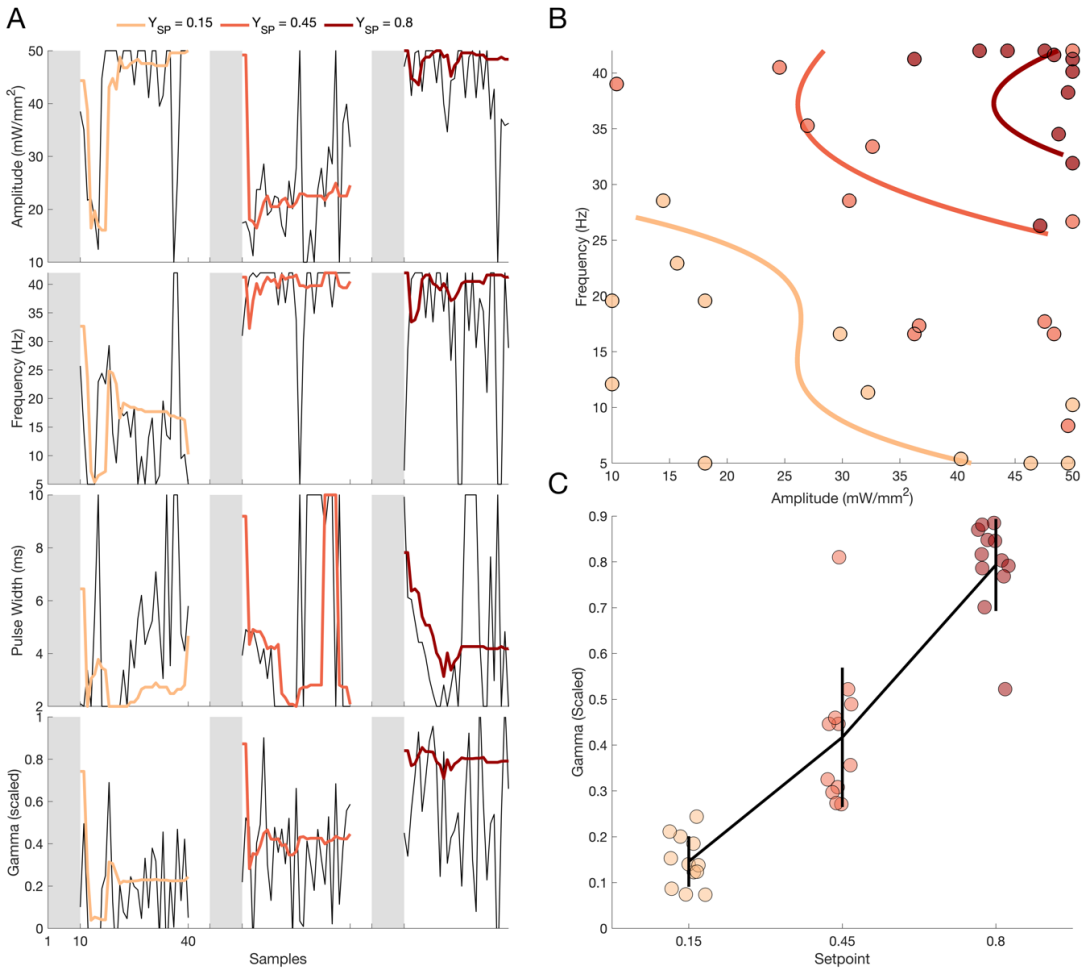

Figure 3.2: Simulation results for setpoint optimization for hippocampal gamma power.

(A) The search trajectories for three representative trials of setpoint optimization. The grey box indicates the 10 burn-in samples. Bold colored lines are the estimated optimal stimulation parameter at each sample. Thin black lines are the actual samples evaluated by the optimization algorithm. (B) The estimated optimal stimulations parameters for each of the setpoints shown in filled circles along with the level set contour lines for each setpoint. (C) The ground-truth effect on hippocampal gamma power according to the model. As in the maximization simulations, the outputs of the model are linearly scaled between 0-1. Each circle represents a trial. Black line and error bars represent the mean and standard deviation across all trials for a given setpoint.

3.3.3 In vivo optimization for setpoint

After prototyping setpoint optimization *in silico*, we then deployed the algorithm *in vivo*. The setpoints used were 1.5, 3, and 20 times the baseline gamma power. **Figure 3.3A** shows the search trajectory for a representative trial in subject R5. As expected from the analysis of the ground-truth models and the simulation experiments, synchronized shifts in the estimated optimal stimulation parameters are seen as the optimization system selects statistically equivalent solutions from the setpoint level set. In this trial, for the larger setpoints the estimated optimal stimulation parameters had higher frequencies, and the estimated optimal amplitudes did not show a specific pattern. While this trial shows a general decrease in the estimated optimal pulse-width as the setpoint increases, this is not consistent between trials/subjects. In **Figure 3.3B**, we show the optimal stimulation parameters estimated for two trials consisting of optimizing for each setpoint in two subjects (R5 and an additional subject R6). The lines connect the three setpoint optimizations conducted in a row as part of one trial within subject. As predicted by the simulation experiments, the two lower setpoints resulted in trials with either high amplitude and low frequency, or vice versa. For the largest setpoint, the optimization is equivalent to a maximization problem, and therefore the estimated optimal stimulation parameters are close together.

We then evaluated the estimated optimal stimulation parameters for each setpoint in a randomized test phase. Results for subject R5 are shown in **Figure 3.3C**. For the first setpoint, 1.5, the estimated optimal stimulation parameters were able to reliably modulate hippocampal gamma power to 1.5 times the pre-experiment baseline. For the next setpoint,

3.0, in one trial the estimated optimal stimulation parameters overshot the desired setpoint, while in the other trial the estimated optimal stimulation parameters undershot the desired setpoint. This mirrors the wider trial-to-trial variability seen with the setpoint of 0.45 in the simulation experiments. For the final setpoint, 20.0, the estimated optimal stimulation parameters were only able to modulate hippocampal gamma to 5.2- and 5.4-times baseline. This underscores the limitation of setpoints to within a physiologically possible range. Finally, **Figure 3.3D** shows the wideband PSD for the estimated optimal stimulation parameters for each setpoint. The PSDs for all three setpoints have peaks within the gamma range (33-55Hz). However, the primary peaks and harmonics are not necessarily aligned outside of the gamma range for the three setpoints as the respective stimulation frequencies were 11.7, 18.4, and 37.9Hz. Focusing on the gamma range in **Figure 3.3E**, there is an increase in power for the frequencies around 37Hz, maximizing the area under this region of the PSD using this constrained stimulation parameter space. Results for additional trials are detailed in 3.5.1.

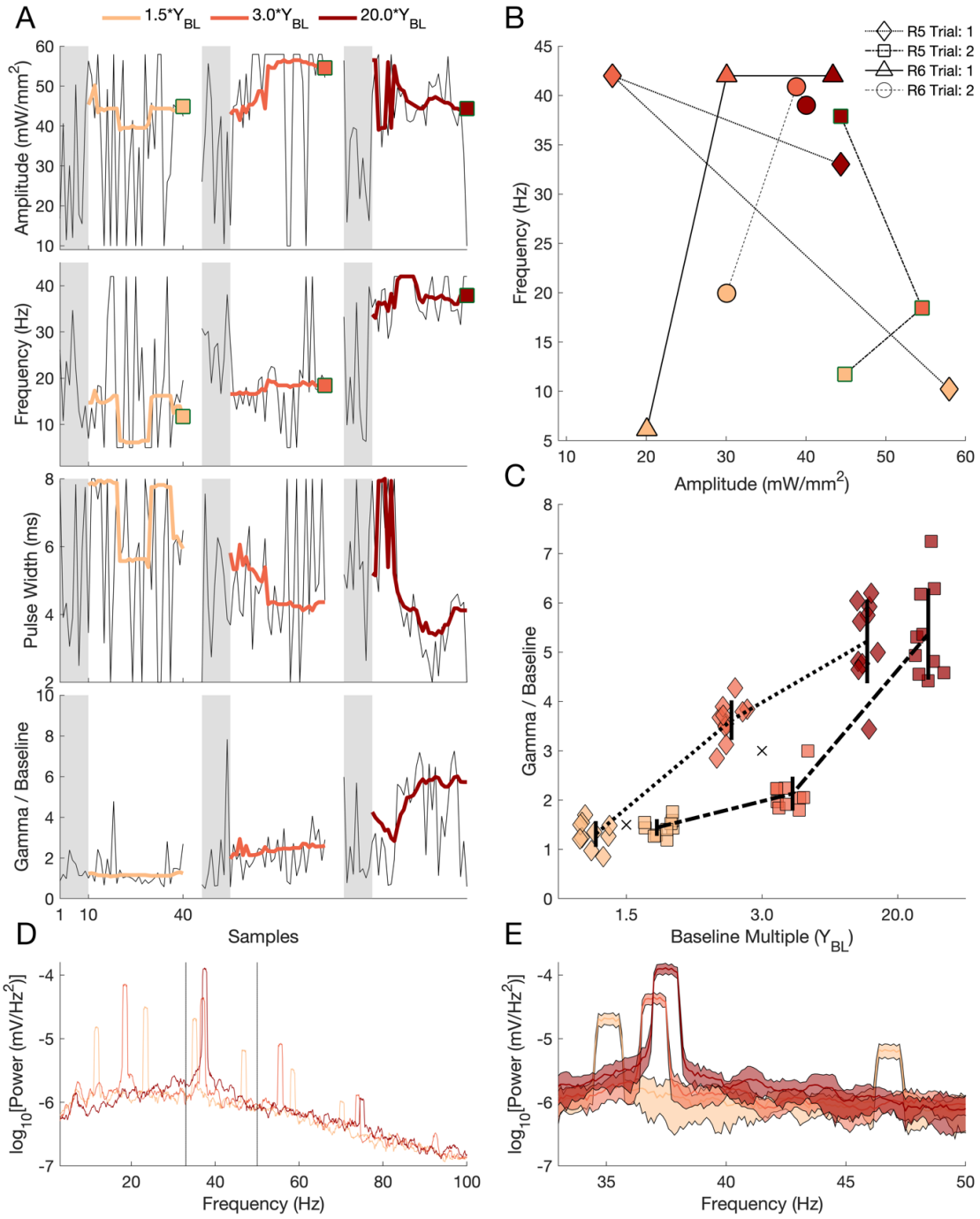

Figure 3.3: In vivo setpoint optimization of hippocampal gamma power.

(A) Search trajectory for a representative trial in subject R5. Grey boxes represent the 10 burn-in samples. Bold colored lines are the estimated optimal stimulation parameter at each

sample. Thin black lines are the actual samples evaluated by the optimization algorithm. The squares at the end of each trajectory for amplitude and frequency correspond to those in Figure 9B. (B) Trajectory of the estimated optimal stimulation parameters across the thee setpoints within a trial for subjects R5 and R6. The estimated optimal parameters for the three setpoints within each trial are connected with a line. (C) Results of the validation-phase for subject R5. Lines connect the three tested stimulation parameters found during each independent experiment session. Vertical black lines show the standard deviation and each marker is an individual sample. 'X' shows the setpoint target for baseline multiples 1.5 and 3. Marker for the baseline multiple of 20 would be off scale. (D) Mean wideband PSD for the test phase of a representative trial. (E) Gamma range PSD. Shaded regions indicate standard deviation across trials.

3.4 Discussion

To test the predictive power of the framework on other objective functions, we adapted the optimization algorithm to instead modulate hippocampal gamma power to a desired setpoint. While the mathematical transformation to create a setpoint objective function was relatively simple, the ground-truth models and the overall simulation platform provided a valuable opportunity to more extensively study the implications of this transformation. In the simulation, the two key findings were, 1) rather than a single optimum there was a level set of stimulation parameters that were effectively equivalent, and 2) the algorithm performed worst on the middle setpoint of the three. Both these characteristics were observed *in vivo*. First, the different stimulation parameters for a given setpoint were markedly different but performed similarly, providing opportunities for additional constraints (e.g. find the stimulation parameters for a setpoint with the lowest energy consumption). Second, by folding the objective function on that setpoint, the overall contrast between the minimum and the maximum was smaller, while the noise remained the same. This results in a lower contrast-to-noise ratio, making the optimization problem more difficult, and is discussed in Chapter 2, Section 5.5.

The extension of the framework to setpoint objectives offers the potential for even more precise brain stimulation therapy and more tightly controlled experimental interventions.

3.5 Supplement

3.5.1 Setpoint results for other experiments

Figure 3.4 shows the search trajectories for the three trials not shown in **Figure 3.3.** These include the first trial for subject R5 and both trials for subject R6. **Figure 3.5** shows the validation phase experiments for subject R6,

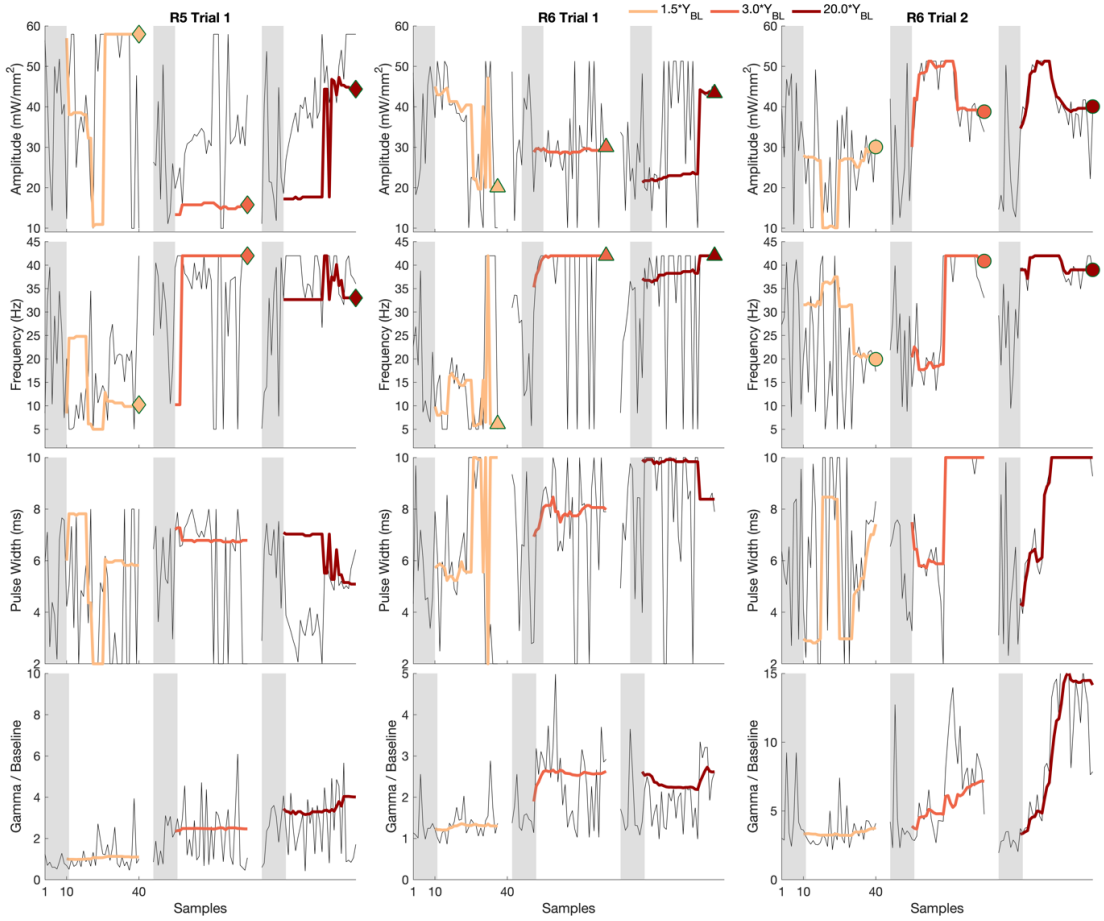

Figure 3.4: In vivo setpoint optimization trajectories.

Search trajectory for a representative trial 1 in subject R5 and trials 1 and 2 for subject R6. Gray boxes represent the 10 burn-in samples. Bold colored lines are the estimated optimal stimulation parameter at each sample. Thin black lines are the actual samples evaluated by the optimization algorithm. The markers at the end of each trajectory for amplitude and frequency correspond to those in figure 6B.

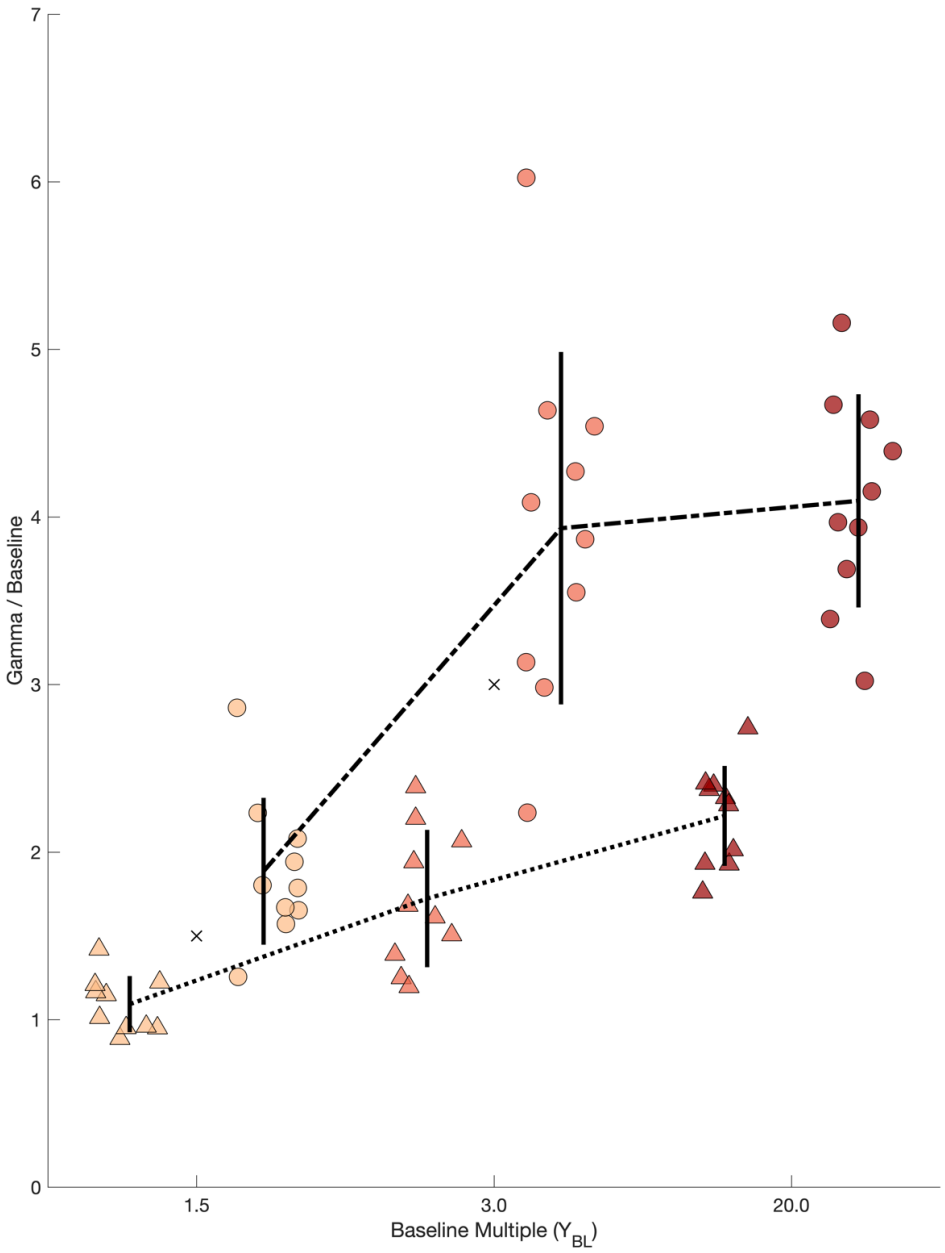

Figure 3.5: Results of the validation-phase for subject R6.

Lines connect the three tested stimulation parameters found during each independent experiment session. Vertical black lines show the standard deviation and each marker is an

individual sample. ‘X’ shows the setpoint target for baseline multiples 1.5 and 3. Marker for baseline the multiple of 20 would be off scale.

CHAPTER 4. APPLICATION OF THE FRAMEWORK FOR STATE-DEPENDENT OBJECTIVES

4.1 Introduction

In Chapters 2 and 3 we considered the response to stimulation to be noisy, but static. In other words, the stimulation setting that is optimal on average does not change. However, this may not always be the case. For example, it is well established that the effect of DBS for PD patients can vary based on the patient's medication level and general physiological state. As a result, the optimal stimulation setting will dependent on the underlying neural state. In order to deliver continually optimal stimulation, the algorithm would need to learn how the effect of stimulation changes with the underlying state, and adjust the stimulation settings accordingly.

To explore how data-driven optimization can be extended to state-dependent objective functions, we construct a model system by delivering medial septum optogenetic stimulation at varying levels of isoflurane anesthesia and measuring hippocampal gamma power. While the mechanism of isoflurane anesthesia is not full elucidated, the general hypothesis is that it decreases the conductance of potassium channels, increasing the refractory period of the neurons. We first investigate how anesthesia alters the response to stimulation, and then propose an extension of Bayesian optimization that learns a state-dependent policy and evaluate it *in silico*.

4.2 Methods

4.2.1 Surgery, virus injection, and electrode implant

Surgical details are as described in Chapter 2, Section 2.1.

4.2.2 Grid search experiment under varying levels of anaesthesia

Neural modulation data was collected under two distinct neural states: while the animal was freely behaving, and while subjects were under 2-3% isoflurane anesthesia. In each condition, a grid search consisting of six frequencies (5, 7, 11, 17, 23, 29, 35 Hz) at six amplitudes (0, 10, 20, 30, 40, 50 mW/mm^2) was conducted. Each stimulation consisted of a pulsatile 4ms square-wave delivered for 4 seconds, and was applied in sequences of 5 stimulations (20s) separated by 40s for washout.

4.2.3 Neural feature calculation

Neural signals from CA3 and CA1 were recorded at 24414Hz and down-sampled to 2 kHz. Each 4 second stimulation epoch (X_1) and 4 second pre-stimulation epoch (X) was segmented along with the corresponding stimulation parameters (U_1). The power spectral density was then estimated using the multi-taper method and summed to measure the overall power for each of the canonical spectral bands (delta: 1-4 Hz, theta: 4-7 Hz, alpha: 8-12 Hz, beta: 13-31 Hz, and gamma: 32-50 Hz). The power in each of these bands is measure of some underlying neural state.

4.2.4 Static GP model fitting

The static Gaussian process was modeled as described in Chapter 2, section 2.4.

4.2.5 State-dependent Gaussian process model

To test optimization approaches on a state-dependent model, we constructed a state-dependent simulation platform from the stimulation and neural state data. The core of the simulation platform is a Gaussian process model mapping pre-stimulation neural state X_0 and stimulation parameters U_1 to the neural state during stimulation X_1. Using this Gaussian process, a set of stimulation parameters U_1 can be sampled to produce a draw from a normal distribution:

$$X_1 = GP(U_1|X_0) = \mathcal{N}(\mu(U_1|X_0), \sigma(U_1|X_0)) \tag{4.1}$$

where μ and σ are the output of the Gaussian process. In this formulation, the neural state during stimulation is a function of both the stimulation parameters and the pre-stimulation neural state

4.2.6 State-dependent Bayesian optimization

For any general optimization problem, the goal is to identify the argument U_1 that maximizes some objective function $F(U_1)$:

$$U_1^* = \underset{U_1}{\operatorname{argmax}} F(U_1) \tag{4.2}$$

Since the objective function is stochastic, we are limited to gradient free optimization methods. Moreover, given the time- and resource-intensive nature of evaluating the effect of stimulation, data-efficient optimization algorithms are well-suited to this problem.

Using the state-dependent simulation platform, we compared two optimization algorithms. The first was the standard implementation of BaO. For an in-depth treatment of BaO, see [44]. Briefly, the BaO algorithm is a model-based, gradient-free optimization algorithm. A small number of burn-in samples are drawn from the stimulation parameter space. The stimulation parameters U_1 and neural state during stimulation X_1 are used to fit a Gaussian process model. An acquisition function is then applied to the Gaussian process which weighs value of sample based on the magnitude of the expectation for and the uncertainty. In this case we used the UCB acquisition function:

$$H_{UCB} = \mu(x) + \sqrt{\kappa_t}\sigma(x) \tag{4.3}$$

where the sequence κ_t was defined as $\kappa_t = 2\log\left(\frac{t^2\pi^2}{6}\right)$, with t being the number of samples collected so far. In other words, the emphasis on exploration increases with each additional sample. The set of stimulation parameters with the highest value for the acquisition function is then sampled. The new set of stimulation parameters and corresponding neural state during stimulation is then used to update the model, and the process is repeated until either stopped or a convergence criterium is met.

By construction, one limitation of the standard implementation of BaO is that it does not take into account state-dependent changes in the objective function it is trying to

optimize. In other words, if the optimal set of stimulation parameters changes as a function of the neural state, BaO is likely to converge to the optimal parameters for the pre-stimulation neural states that more prevalent, converge to a set of parameters that is a weighted average of the different optima across states, or not converge at all.

To address the limitations of the BaO algorithm we extended the algorithm to not just learn the effect of the stimulation, but to take into account the pre-stimulation state when stimulation is applied. This approach, state-dependent Bayesian optimization (SDBO), still uses a Gaussian process to approximate the objective function which maps the stimulation parameters U_1 *and* the pre-stimulation neural state X_0 to the neural state during stimulation. During the learning phase, SDBO

4.2.7 Simulation experiments

Both optimization approaches, BaO and SDBO, were evaluated in a two-phase experiment. In the first training phase, the algorithms interact with the simulation platform by sampling stimulation parameters, measuring their effect on neural state, and updating their models. In the test phase, the optimization algorithms apply their learned policies without exploration to maximize the neural state in the simulation platform. Both phases consist of 100 interactions with the simulation platform, and each train-test cycle is repeated 30 times to characterize the general behavior of the algorithm.

4.3 Results

4.3.1 Effect of stimulation changes under isoflurane anaesthesia

Comparing the cross-validated NMSE across all three models characterized the contribution of the stimulation parameters, the pre-stimulation neural state, and the combination of both to predicting the during-stimulation neural state. Based on this analysis we found that different spectral bands have different levels of state- vs. stimulation-dependent effects (**Figure 4.1**). For the band with the lowest frequency, delta (1-4 Hz), the static model (U_1) performed worse than both the state (X_0) and joint ([U_1, X_0]) models, indicating that the pre-stimulation state is the best predictor of the during-stimulation state, and that stimulation has little to no effect. Both theta (4-7 Hz) and alpha (8-15 Hz) did not have NMSEs that were significantly lower than 1, indicating that neither the stimulation parameters (U_1) or the pre-stimulation state (X_0) were significant predictors of during-stimulation state (X_1). In contrast to the previously mentioned spectral bands, for beta power (16-31 Hz), the static model and the joint model both had significantly lower NMSE than the model with state alone. This suggests that the best predictor of during-stimulation beta power is the stimulation parameters. Finally, for gamma power, both the static model and the state model had similar NMSEs, while the joint model was an improvement over both. This finding suggests that the effect of stimulation parameters on during-stimulation neural state is directly influenced by the pre-stimulation neural state. Based on these findings, successful optimization of gamma power requires adjusting stimulation parameters based on the current neural state.

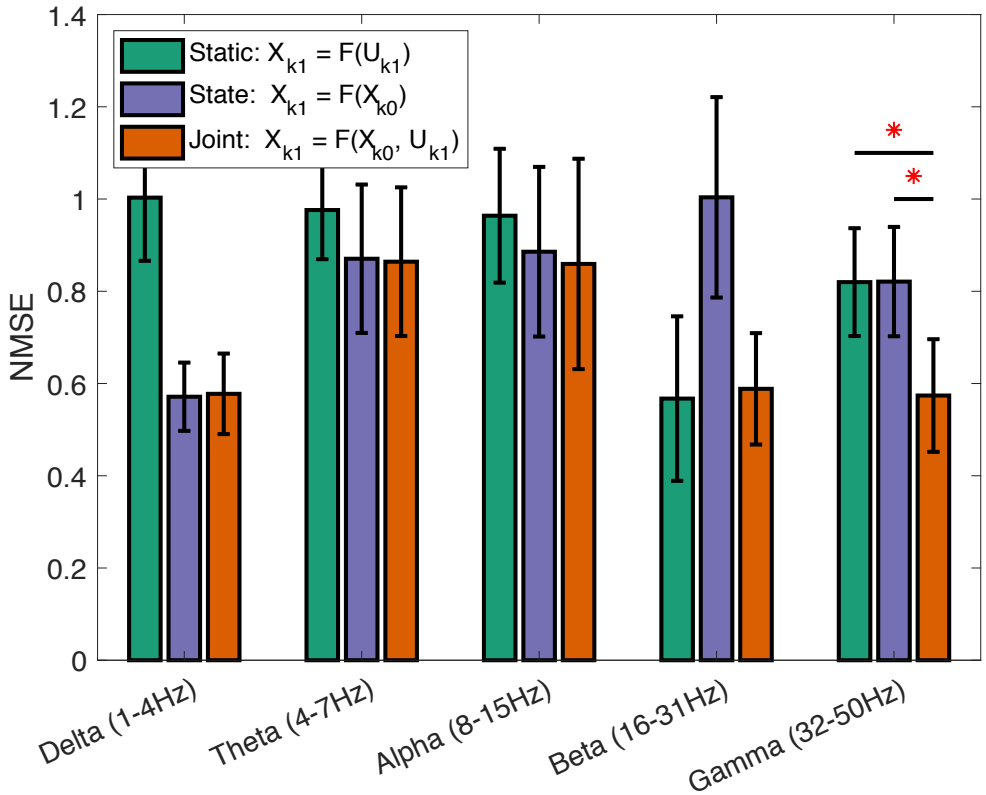

Figure 4.1: Gaussian process regression model accuracy for stimulation parameters, baseline state, and joint predictors.

Each set of bars indicates the mean NMSE across 20-folds of cross-validation ± standard error. Red stars over gamma show the set of significance tests (t-test, $p < 0.05$) that demonstrate hippocampal gamma is dependent on stimulation parameters and pre-stimulation state.

4.3.2 Modelling identifies gamma power as dependent on anaesthetic levels

The change in how in the mapping from stimulation parameters to during-stimulation state based on pre-stimulation state is demonstrated in **Figure 4.2**. This figure shows the mapping from stimulation parameters to during-stimulation gamma power is altered as gamma increased from the lowest value observed during isoflurane anesthesia to the highest value observed while the subject is freely behaving. Initially, at the lowest pre-stimulation gamma power, the stimulation parameters that maximize the during-

stimulation gamma power are 35 Hz at 50 mW/mm^2. However, as pre-stimulation gamma power increases, a local maximum forms around 15 Hz at a lower amplitude of 25 mW/mm^2. As the pre-stimulation gamma approaches the highest value observed in the freely behaving subjects, this local maximum becomes the global maximum. Tracking the progression of the stimulation parameters corresponding to this global max demonstrates that an optimization algorithm to maximize gamma power should take into account the pre-stimulation gamma neural state.

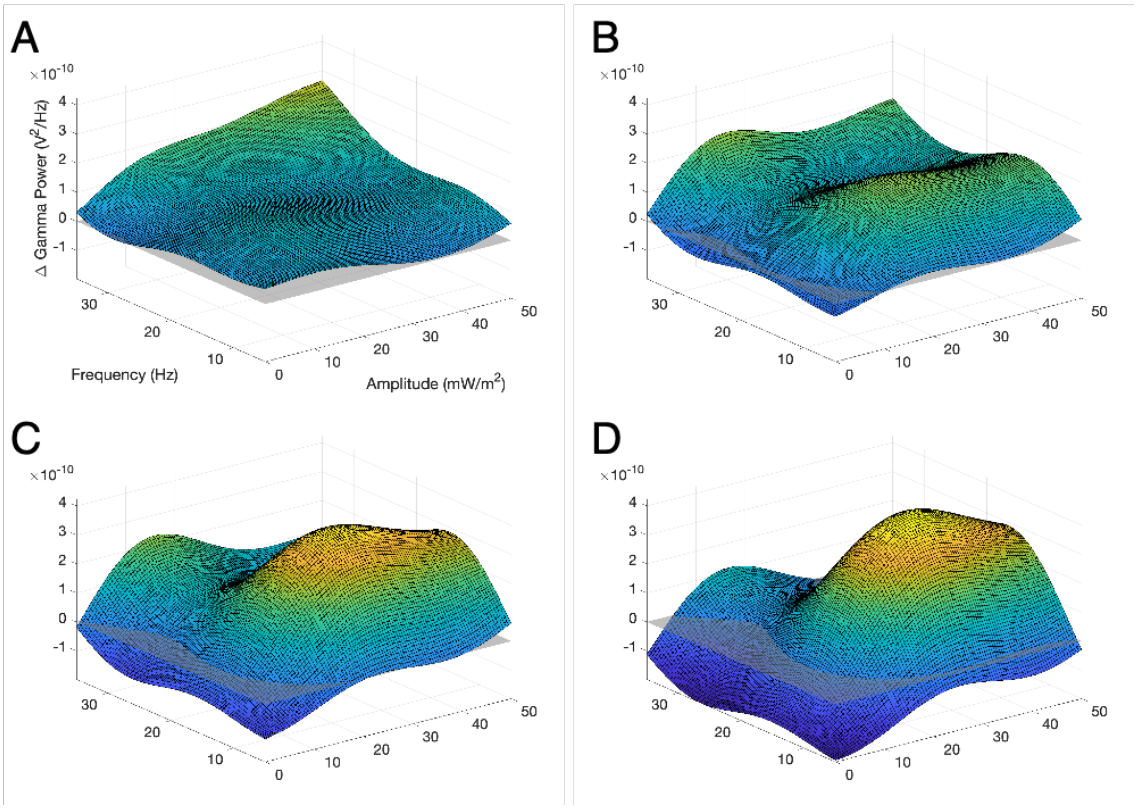

Figure 4.2: Change in gamma at different values of baseline gamma power.

(A-D) Each surface is a cross-section showing the effect of stimulation on the change in hippocampal gamma at different levels of baseline gamma power. Baseline gamma increases from the lowest value observed (A) to the highest value observed (D).

4.3.3 Using framework to prototype state-dependent Bayesian optimization

Both the BaO and SDBO algorithm were applied to the platform which simulates the stochastic effect of stimulation parameters on during-stimulation state, but changes the effect of the stimulation parameters based on the pre-stimulation state. **Figure 4.3A** shows the median and quantiles of the during-stimulation gamma power for both algorithms. Initially, when the pre-stimulation state is low (~0-1.5x10^{-10} V^2/Hz) the policies learned by both algorithms perform comparably well. However, once the pre-stimulation state rises above 1.5x10^{-10} V^2/Hz, SDBO performs substantially better. In other words, the parameters selected by SDBO are able to induce a higher during-stimulation gamma state than the single set of stimulation parameters learned by BaO. As the pre-stimulation neural state then decreases, both algorithms again perform similarly.

Figure 4.3B compares the policies learned by both algorithms to the optimal ground-truth policy from the simulation platform. The optimal policy (black line) shows that for low gamma pre-stimulation states, the stimulation parameters that most increase gamma are at 50 I and 35 Hz. However, as the pre-stimulation gamma transitions from 1.5x10^{-10} V^2/Hz and 5x10^{-10} V^2/Hz the stimulation parameters that most increase gamma during stimulation are around and intensity of 25 mW/mm^2 and a pulse frequency of 15 Hz. While the policy learned by SDBO tracks the change in optimal parameters (orange line), BaO selects a single parameter that is closer to the optimal for the states with lower gamma power, but typically falls in between the optimal parameters for lower- and higher-gamma power states.

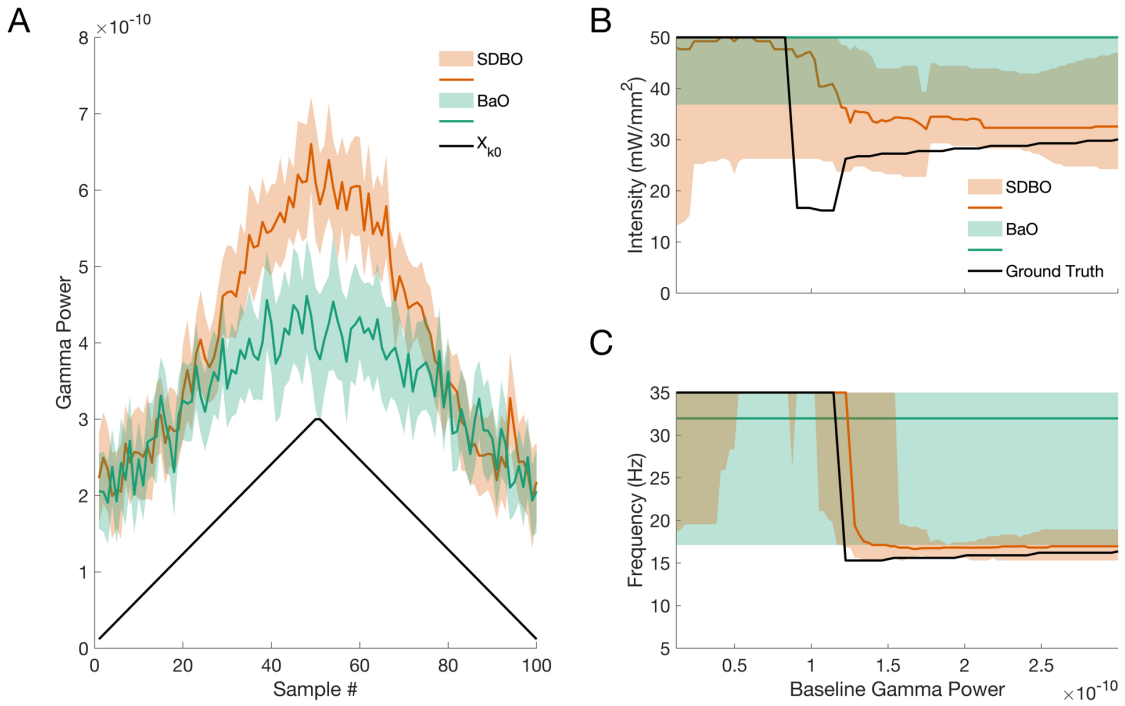

Figure 4.3: Performance and behavior of BaO and SDBO during test phase in silico. (A) The during-stimulation gamma power for BaO (green) and SDBO (orange), patches indicating the quartiles across 30 trials. Black line shows the baseline gamma power before stimulation. (B-C) The distributions of static parameters learned by BaO, and the policies learned by SDBO. Black line is the optimal policy from the ground truth model.

4.4 Discussion

In this study, we found that the mapping from stimulation parameters to gamma state during stimulation is dependent on a pre-stimulation gamma neural state. We then created a platform that simulated the effect of medial septum optogenetic stimulation on hippocampal gamma power, based on a changing pre-stimulation gamma state. Using this platform, we then compared two algorithms for optimizing stimulation parameters to maximize gamma state: the standard implementation of static Bayesian optimization, and a state-dependent extension that improved performance.

4.4.1 Stimulation parameters are not one-size-fits-all-states

It is a reasonable hypothesis that the effect of applying stimulation to a neural system is going to be dependent on not only the characteristics of the stimulation applied, but also the current state of the neural system. When we tested this hypothesis in our medial septum optogenetic stimulation model we found that this was not always the case. For some spectral bands used as measures of neural state, the prediction of the next state X_1 is entirely dependent on: the current state X_0 (delta), the stimulation parameters U_1 (alpha), both (gamma), or neither (theta and beta). While it would have been interesting had all biomarkers been state-dependent within our parameters space, the approach we used of comparing the NMSE of GP regression models is relatively non-conservative. Specifically, if the change in pre-stimulation neural state causes a shift in the mapping from stimulation parameters to neural state without changing the location of the global extrema it would still register as state-dependent. In the context of optimization, this would not be relevant as the optimal parameters would not change. However, this could play a role when adapting this approach to control around a setpoint.

4.4.2 Framework can be extended to state-dependent objectives

Using our simulation model of the state-dependent effect of medial septum optogenetic stimulation we tested standard Bayesian optimization (BaO) and a state-dependent extension (SDBO). We found that SDBO out-performed BaO when the mapping from stimulation parameters to gamma state is changing. The BaO algorithm was limited because it would use all previously collected samples to estimate the optimum, selecting a

policy based on the weighted average of states and their optimal parameters. i.e., the learned policy is between the two true optimum, but closer to the optimum for the states more encountered. In contrast, SDBO learns a state-dependent policy that does not conflate optima at different states. This approach could be further extended to use additional measures of state to further improve the learned policy.

CHAPTER 5. APPLICATION OF THE FRAMEWORK FOR SAFETY CRITICAL OPTIMIZATION

5.1 Introduction

While neural modulation can be an effective therapy, it can also cause side effects. For optimization to be used in clinical practice, it needs to account for subject- specific safety considerations. While DBS is generally well-tolerated in patients, regions of the parameter space that are optimal in some subjects can produce unwanted and uncomfortable side-effects in other subjects. These include muscle contractions [55], voice changes [56], and abnormal sensations. While side effects or returning symptoms caused by a change to the stimulation setting may be briefly tolerable under the supervision of a clinician, they could be unsafe or detrimental during daily living activities [57]. As a consequence, it will be critical to design data-driven that can actively learn and avoid unsafe parameter space regions for any automated or unsupervised optimization approaches. This can be accomplished by building on recent work demonstrating how learnable safety constraints can be integrated with a Bayesian optimization approach to safely and efficiently navigate a parameter space [58].

Recent work in the field of optimization has produced algorithms specifically designed for safety-critical optimization [59]. Many systems in the aviation and industry require a tuning and optimization process to achieve optimal performance. However, if the systems are misconfigured, the results could be catastrophic. While safety-critical

optimization methods can be implemented across a range of algorithm types (e.g., evolutionary or gradient approximation) we limit our focus to Bayesian optimization approaches. In the context of neural modulation, a safety-critical optimization algorithm has three goals, exploitation – finding the best stimulation setting, exploration – testing stimulation settings with high uncertainty, and safety – avoiding regions of the parameter space that have a high likelihood of being unsafe. The exploitation and exploration are based on the surrogate model of the objective, as in Chapters 2-4, while the safety can be based on the same surrogate model, a model of a specific safety metric, or multiple safety metrics.

This chapter extends the previously described framework [37] to prototype and deploy a Bayesian optimization algorithm with learnable safety constraints called META-SAFE (**Figure 5.1**). We designed, characterized, and deployed this algorithm in normal rats undergoing a spatial object recognition (SOR) memory task while receiving hippocampal stimulation. The stimulation paradigm used was originally developed to reduce seizures in the rat tetanus toxin model of temporal lobe epilepsy [60], and was later shown to impair performance on the SOR task at sufficiently high stimulation amplitudes [61]. Using this model system, we designed and validated an optimization algorithm with the goal of efficiently maximizing the stimulation while carefully avoiding regions of the parameter space that could cause memory deficits.

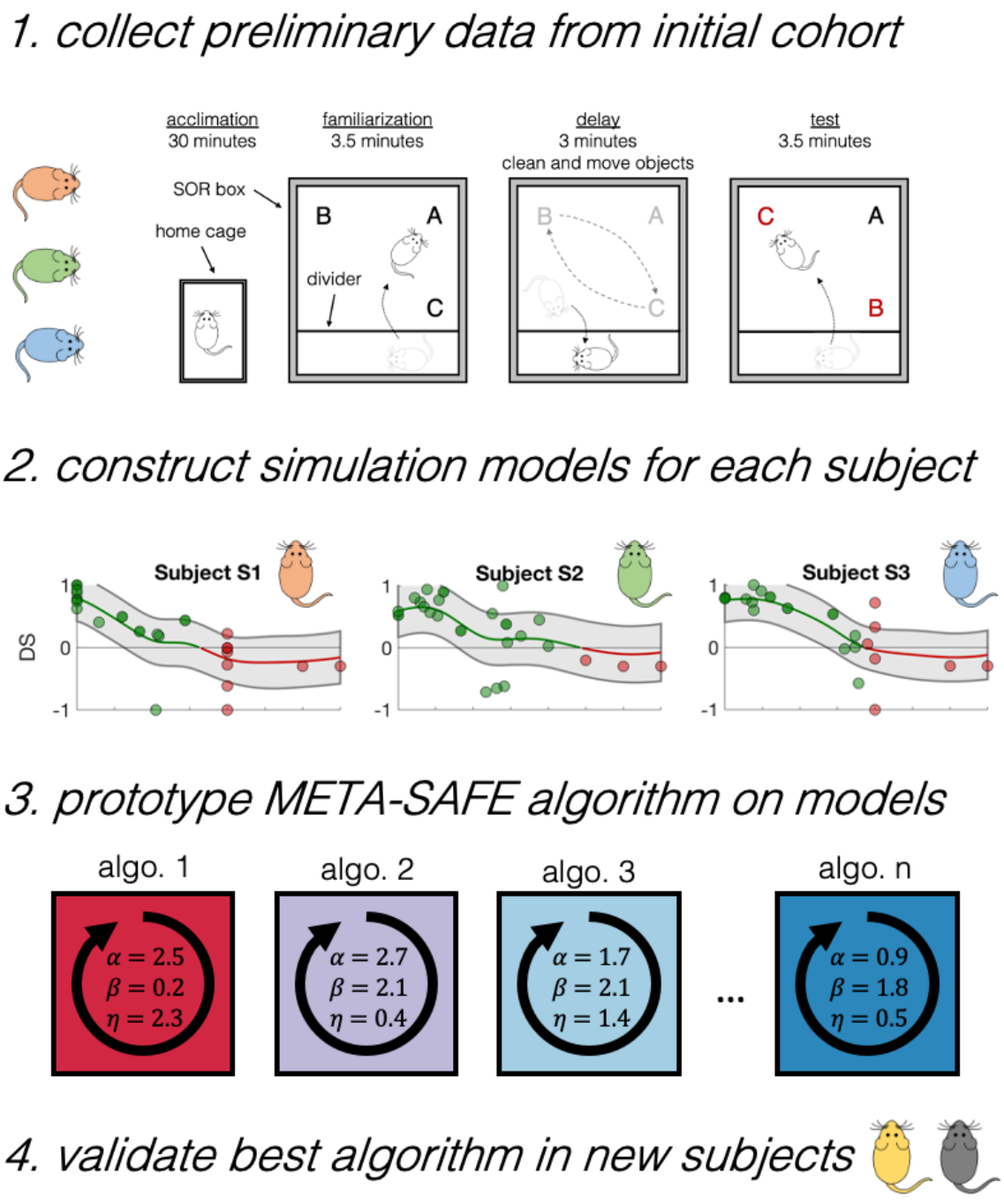

Figure 5.1: META-SAFE design framework.

The design framework was applied in four phases. 1) Data was collected from an initial cohort of subjects that approximate the diversity in the general population by applying different stimulation settings and measuring the resultant performance on the memory task. 2) The data from each subject was used to create a simulation model that, on a statistical

level, could act as a proxy for the original subject, but with advantage that testing a stimulation *in silico* was nearly instant relative to *in vivo*. 3) Instead of prototyping and refining the META-SAFE algorithm on real subjects, the different configurations of META-SAFE were rapidly evaluated on each of the proxy simulation models. 4) The configuration of META-SAFE that performed best across the proxy simulation models was then validated *in vivo* in a new cohort of subjects.

5.2 Methods

The design framework was applied in four phases. 1) Data was collected from an initial cohort of subjects that approximate the diversity in the general population by applying different stimulation settings and measuring the resultant performance on the memory task. 2) The data from each subject was used to create a simulation model that, on a statistical level, could act as a proxy for the original subject, but with advantage that testing a stimulation *in silico* was nearly instant relative to *in vivo*. 3) Instead of prototyping and refining the META-SAFE algorithm on real subjects, the different configurations of META-SAFE were rapidly evaluated on each of the proxy simulation models. 4) The configuration of META-SAFE that performed best across the proxy simulation models was then validated *in vivo* in a new cohort of subjects.

5.2.1 Data collection

5.2.1.1 Subjects and surgery

Subjects underwent a single survival surgery to implant a 16-channel MEA in the CA3 and CA1 regions of the hippocampus. Implantation details are identical to those in the second survival surgery described in Chapter 2, Section 2.1, with the omission of the fiber optic ferrule in the medial septum.

5.2.1.2 Asynchronous distributed stimulation

The stimulation pattern consisted of symmetric square wave biphasic pulses with 400 μs per phase on eight stimulating contacts [60]. Stimulation was applied fully asynchronously where each contact had a pulse frequency of 7 Hz with a phase offset of $\frac{1}{7*8}s$. Stimulation was voltage controlled with amplitudes ranging from ±0-5 V.

5.2.1.3 Spatial object recognition (SOR) task

The SOR task is a well-established behavioral task to test object recognition memory in various animal species [62]. The task relies on the innate preference of rats for novel objects and changes in object location. Before starting the SOR experiments, the implanted rats were habituated to a 2'x1.5' open-field SOR box for 5 minutes per day for three days. Practice objects were placed in the SOR box on the third day to determine whether the subjects were willing to explore the objects or additional habituation was necessary.

Each SOR task consists of four phases: acclimation, familiarization, delay, test. (**Figure 2**). In the acclimation phase, the rat was moved to the testing room 30 minutes before the SOR task is performed. Meanwhile the SOR box was thoroughly cleaned with 70% isopropyl alcohol to remove any olfactory signals, and three distinct objects were placed in the corners of the exploration area. At the end of the acclimation phase, the headstage was connected to the electrode on the subject, who was then moved to a region of the SOR box separated from the exploration area by an opaque divider. The headstage was connected to a rotating commutator suspended above the SOR box and stimulation

was started. After 1 minute of stimulation, the divider was temporarily removed to begin the familiarization phase. The subject was allowed to interact with the objects for 3.5 minutes, before being again separated from the exploration area by the divider for 3 minutes. During this delay phase, the exploration area and the objects were thoroughly cleaned with 70% isopropyl alcohol to remove any olfactory signals. During the testing phase, the subject was returned to the exploration area and allowed to explore the objects for 3.5 minutes. The rat was then removed from the exploration area to complete the task

The orientation of interchanged objects was counterbalanced among the three corners to prevent any bias. A camera placed on the top of the SOR box was used to capture the video of the familiarization and the testing phases. SOR experiments were performed in a sound-isolated room with limited red light.

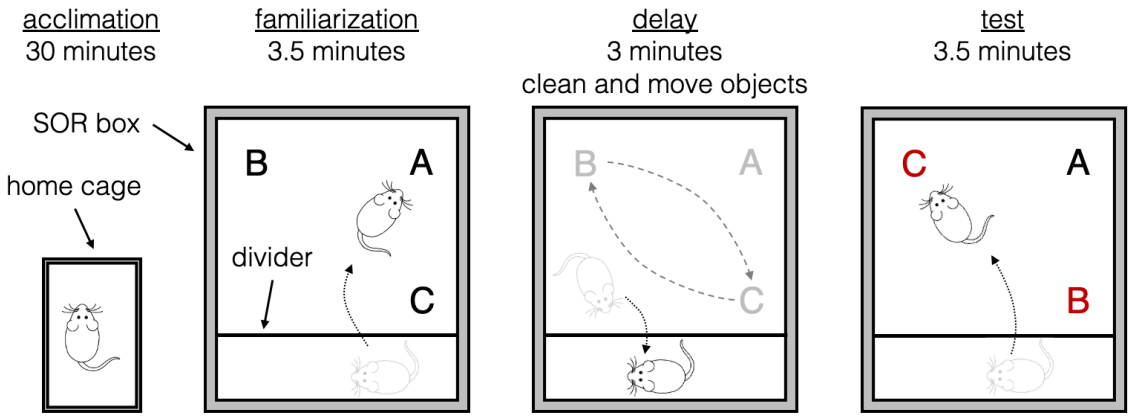

Figure 5.2: Overview of the SOR task.

Starting with the habituation phase, the subject is allowed to acclimate to the room in their home cage for 30 minutes. The subject is then connected to the headstage and placed in the SOR box, separated from the objects by an opaque removable divider. In the familiarization phase, stimulation is started and after 1 minute the divider is temporarily removed to allow the subject to interact with the three objects labeled A, B, and C. After 3.5 minutes the subject is again separated from the field by the divider. The field and objects are cleaned,

and the objects are returned with two of them in swapped positions. After 3 minutes, the divider is again removed for the subject to interact with the objects for 3.5 minutes. Each SOR task produces a single DS measurement.

5.2.1.4 Discriminant score

Discrimination score (DS) was used to quantify the animal's memory performance. DS is defined as the difference in exploration time for interchanged objects and exploration time for stationary object divided by the sum of exploration time for interchanged objects and exploration time for stationary object:

$$\frac{\boldsymbol{mean}(\boldsymbol{T_B}, \boldsymbol{T_C}) - \boldsymbol{T_A}}{\boldsymbol{mean}(\boldsymbol{T_B}, \boldsymbol{T_C}) + \boldsymbol{T_A}} \tag{5.1}$$

Where $\boldsymbol{T_A}, \boldsymbol{T_B}$ and $\boldsymbol{T_C}$, represent the time the subject spent interacting with the three objects. The score ranges from -1 to +1. A score of +1 indicates preference only for the interchanged objects and a score of -1 indicates preferance only for the stationary object. A score of 0 suggests equal preferenence for all three objects and is indicative of no memory. Exploration of an object was defined as the orientation of animals's snout directed towards the object with the whiskers actively twitching, which suggests sniffing. Any other interactions such as running around the object, playing with the object, sitting or climbing on the object was not recorded as exploration. Videos of the SOR task were scored by a blinded observer using Spike2 (Cambridge Electronic Design Limited, Cambridge, England) to annotate the start and stop of each interaction with an object.

Only one sample was collected per day to prevent any interaction between sequential stimulations or decreases in performance due to fatigue. SOR tasks where the

headstage was disconnected, an object fell, or the experiment was otherwise interupped were discarded. SOR tasks where stimulation caused an adverse reaction: electrographic afterdischarges, seizures, etc. were stopped and scored with a DS of -1 to discourage the algorithm from further sampling from that region of the parameter space.

5.2.1.5 Collected data and data augmentation

The data used in the design framework was collected from grid and optimization experiments for 4 subjects conducted as part of a previous study and the 2 subjects included in the standard Bayesian optimization *in vivo* experiments in Chapter 5, Section 3.4.1. For some subjects, the maximum amplitude tested (±4 V) did not disrupt memory, i.e., the average DS did not drop below 0. In these cases, the dataset was augmented with artificial data points at ±6 and ±7 V, to create a region of the parameter space with an average DS of approximately 0.

5.2.2 Ground-truth models for simulating optimization

5.2.2.1 Ground-truth models of DS

For each subject, the input stimulation amplitude and corresponding output DS were fit to a Gaussian process (GP) regression model [45] as described in [37], [63]. The GP was composed of mean and covariance functions:

$$\mu = m(\mathbf{x}') \tag{5.2}$$

$$\Sigma = \mathrm{k}(\mathbf{x}, \mathbf{x}') \tag{5.3}$$

where x is a vector of the data used to construct the model, and x' is an unlabeled sample point. In this application the input to the model was a single dimensional parameter space representing the stimulation amplitude discretized on ±0.05 V intervals. The output of the model was the DS drawn from a Gaussian distribution.

$$DS \sim \mathcal{N}(\mu, \Sigma) \tag{5.4}$$

The GP was a constant mean function and a third order Matérn kernel with automatic relevance determination for the covariance:

$$k(x_i, x_j) = \sigma_f^2\left(1 + \sqrt{3}r\right)\exp\left(-\sqrt{3}r\right) \tag{5.5}$$

where σ_f^2 is the variance of the output and:

$$r = \sqrt{\sum_{m=1}^{d} \frac{\left(x_{im} - x_{jm}\right)^2}{\sigma_m^2}} \tag{5.6}$$

where σ_m^2 is the length scale hyperparameter describe the variance of the stimulation amplitude parameters, x_m, [45]. In other words, σ_m^2 represents how much the expectation can vary between adjacent stimulation amplitudes. Each model was individually reviewed and model hyperparameters (mean, length scale, signal noise, measurement noise) were manually adjusted in the event of overfitting or non-physiological responses.

5.2.2.2 Discrimination area (DA) objective function

In previous studies, it was shown that optimizing stimulation amplitude based on DS produced a trivial optimum close to 0.0 V – i.e., no stimulation. To address this, the objective function used in this study was the discrimination area, or the DS multiplied by the stimulation amplitude. By construction, this objective function possessed several convenient properties. Specifically, when either the amplitude or DS were equal to zero, the DA was also zero. However, at a midpoint when a stimulation with an amplitude greater than zero is applied and does not completely disrupt memory, the DA is positive. Therefore, since the Matérn kernel of the GP is differentiable, according to Rolle's theorem, there must exist at least one stimulation amplitude for which the derivative is zero, which based on previous data. is likely a non-trivial maximum of the objective function.

5.2.3 Meta-Bayesian optimization with learnable safety constraints

5.2.3.1 Bayesian optimization

The Bayesian optimization algorithm used in this study was based on a GP model and an upper confidence bound (UCB) acquisition function as described in [44]. The algorithm initializes by sequentially applying a set of burn-in stimulation settings (amplitude). The objective function, DA, is calculated and the data is fit to a surrogate GP model. The acquisition function was then used to quickly identify regions of the parameter space where the model predicts the average DA would be high (exploitation), but also had a large uncertainty with the statistical potential for even higher DA value (exploration). The UCB acquisition function was defined as:

$$H_{UCB} = \mu_{DA}(x) + \sqrt{\eta\kappa_t}\sigma_{DA}(x) \quad (5.7)$$

Where the configuration parameter, η, scaled how much to weigh exploitation vs. exploration, the sequence κ_t was defined as:

$$\kappa_t = 2\log\left(\frac{t^2\pi^2}{6}\right) \quad (5.8)$$

where t was the number of samples.

5.2.3.2 Bayesian optimization with learnable safety constraints

To implment Bayesian optimization with learnable safety constraints, we extended the UCB acquisition function with the additional safety criteria designed to avoid stimulation settings expected to produce an average DS less than zero. When selecting a stimulation setting, the surrogate DS model was queried to identify regions of the parameter space that satisfied the criteria:

$$\mu_{DS}(x) - \beta * \sigma_{DS}(x) > 0 \quad (5.9)$$

where β was a configurable parameter to control how carefully the algorithm would avoid potentially unsafe regions of the parameter space. A high β would cause the algorithm to avoid stimulation settings with even a small risk of producing a mean DS below zero. Wherease, a β close or equal to zero, would cause the algorithm to only avoid regions of the parameter space where it expected the mean DS to zero or lower.

5.2.3.3 Bayesian optimization with hyperpriors

The algorithm was further extended to incorporate information from prior subjects in the form of hyperpriors which could further improve the performance of the algorithm [45], [63]. When fitting the model by maximizing the likelihood over the space of hyperparameters defined in Chapter 5, Section 2.2, a prior distribution was placed on the hyperparameters. This distribution was calculated based on the ground-truth models for each *other* subject. The GP kernel used in this study has hyperparameters for the mean function, the length scale for the stimulation amplitude parameter, the signal noise, and the measurment noise. For each of the hyperparameters, the mean and variance across subjects was calculated and used to define a Gaussian distribution hyperprior for that hyperparameter. When fitting the model during optimization, the Gaussian distribution hyperprior was defined for each of the hyperparameters using these statistics. A configuration parameter, α, was multiplied the variance of the hyperprior, such that a small α would emphasize hyperparameters that are close to the hyperprior mean, while a large α would allow a broader range of hyperparameters to be considered. After each additional stimulation setting was tested, the model was re-fit using the hyperpriors and the entire parameter space re-estimated to compute the acquisition and safety functions form the surrogate model.

5.2.4 Simulation experiments

In silico experiments were performed using the ground-truth simulation models. In an initial component test, four experiments were run using the combinations of Bayesian

optimization, Bayesian optimization with learnable safety constraints, Bayesian optimization with hyperpriors, and Bayesian optimization with learnable safety constraints and hyperpriors. The configuration parameters for the algorithms were selected to highlight the characteristic behaviors of the different optimization approaches.

In a second experiment, the configuration parameters for Bayesian optimization with learnable safety constraints and hyperpriors were more fully characterized. The configuration parameters were selected from uniform distributions of η (exploration) = [0 4], β (safety) = [0 3], and α (hyperprior) = [0 3]. For each of the six subjects, the same 30 combinations of configuration parameters were evaluated for 15 trials of 15 samples each.

5.2.5 Algorithm performance criteria

The algorithms were evaluated based on two criteria: final error and overshoot. Final error was calculated as the difference between the DA produced by the estimated optimal stimulation amplitude when applied to the ground-truth model and the max of the ground-truth model. The overshoot was calculated as the difference between the maximum stimulation amplitude and the estimated optimal stimulation amplitude.

5.2.6 In vivo optimization

5.2.6.1 Optimization phase

Each optimization trial was initialized with a set of presumed safe stimulation settings: ±0.0, 0.5, and 1.0 V. The DS was scored for each SOR task and used to fit the surrogate GP model. Optimization continued until the estimated optimal stimulation setting

remained the same for three samples, or 30 samples total. During optimization, data from all trials were included, even if the subject did not interact with each object in the test phase of the SOR task.

5.2.6.2 Validation phase

In the validation phase, we prospectively randomized stimulation between ±0V/sham, the optimal stimulation setting estimated by the algorithm, x*, and a control point specified as the lower of either: $x^* + 1V$ or $2 \cdot x^*$. If the control point did not produce a memory deficit in a given subject, an additional control point at $x^* + 2V$ was also evaluated.

5.3 Results

5.3.1 Data collection

Stimulation and memory performance data was collected from six male Sprague Dawley rats between 16-50 weeks of age. The data from each subject included the stimulation amplitude and DS for the corresponding SOR task. Stimulation amplitudes included were based on a combination of previous optimization and system identification experiments.

5.3.2 Ground truth models

Ground-truth models were constructed using all available data for each of the six subjects. The stimulation amplitudes ranged from ±0-5 V, with two additional artificial data points at ±6 and 7 V (**Figure 5.3**). The estimated optimal stimulation amplitudes ranged from ±1.1 V to ±4.0 V, with safety thresholds ranging from ±1.8 V to ±4.8 V. In all

subjects, once the expectation of the simulation model crossed safety threshold it did not rise back above at higher stimulation amplitudes.

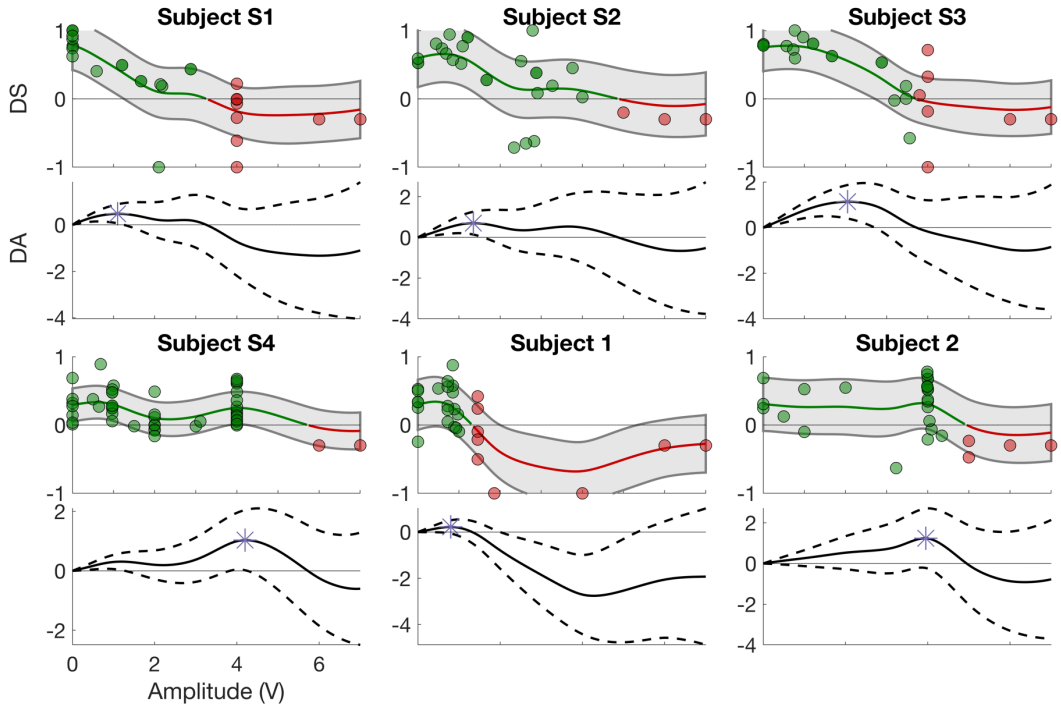

Figure 5.3: Ground-truth simulation models.

The simulation model mapping stimulation amplitude to the DS and the corresponding DA objective function. In the DS plot the green line shows the region of the ground-truth model where the expected DS is greater than 0, while the red portion of the line is when the expected DS is less than zero. The green and red circles show the data used to construct the ground-truth models when the DS measurement was above and below zero, respectively. The gray patch shows the standard deviation. In the DA plot, the solid and dashed lines indicate the expected DA based on the DS model and the standard deviation, respectively. The purple star indicates the estimated optimal of the DA objective function.

5.3.3 Simulation experiments

We first broadly characterized the behavior of the four different versions of the safe-opt algorithm in a representative subject (**Figure 5.4**). As expected, in both the approaches without learnable safety constraints, the Bayesian optimization algorithm

selected simulation amplitudes at the high end of the parameter space that were below the safety threshold. When learnable safety constraints are employed, the search algorithm successfully avoided stimulation amplitudes above the safety threshold. However, when the hyperprior is not used, the search strategy is more conservative and does not fully explore the entire safe regions of the parameter space. When both components are employed, the search strategy fully explores the safe region of the parameter space without sampling from the unsafe regions.

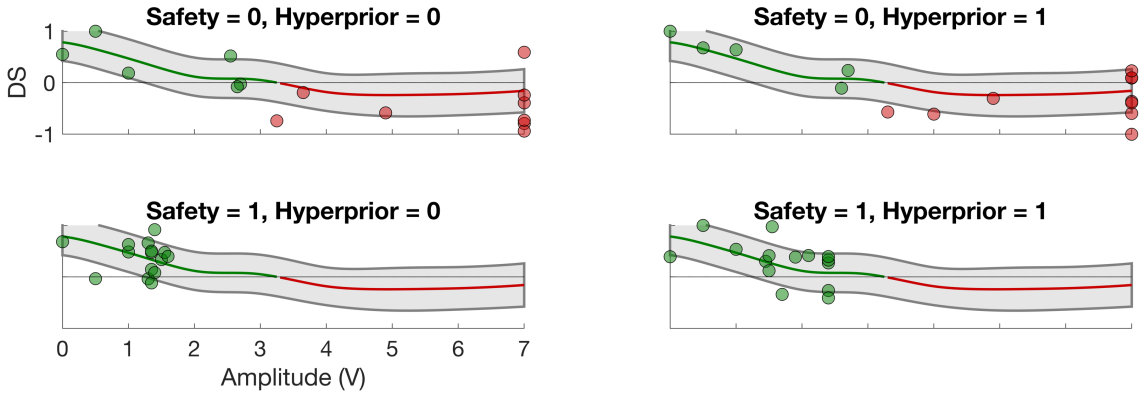

Figure 5.4: Component test.

Simulated results using the four possible combinations of learnable safety constraints and hyperpriors on a representative subject model. The green and red lines represent the expected DS from the ground truth models for the safe and unsafe regions of the parameter space, respectively. Each circle represents the data collected during a simulated optimization trial.

The influence of configuration parameters on the performance of the full Bayesian optimization with learnable safety constraints and hyperpriors was then quantified (**Figure 5.5**). The best, average, and worst performance of each of the 30 combinations of the three configuration parameters was compared on the ground-truth models. Across configurations there was little influence on the best and average final error, though more variability for

the worst final error. However, the overshoot was sensitive to changes in the configuration settings, specifically β (**Figure 5.5B**). As β is increased, the average and worst case overshoot significantly decreases (p-value < 0.01), indicating a more conservative search strategy. The other two parameters, η and α were not significant predictors of either performance metric.

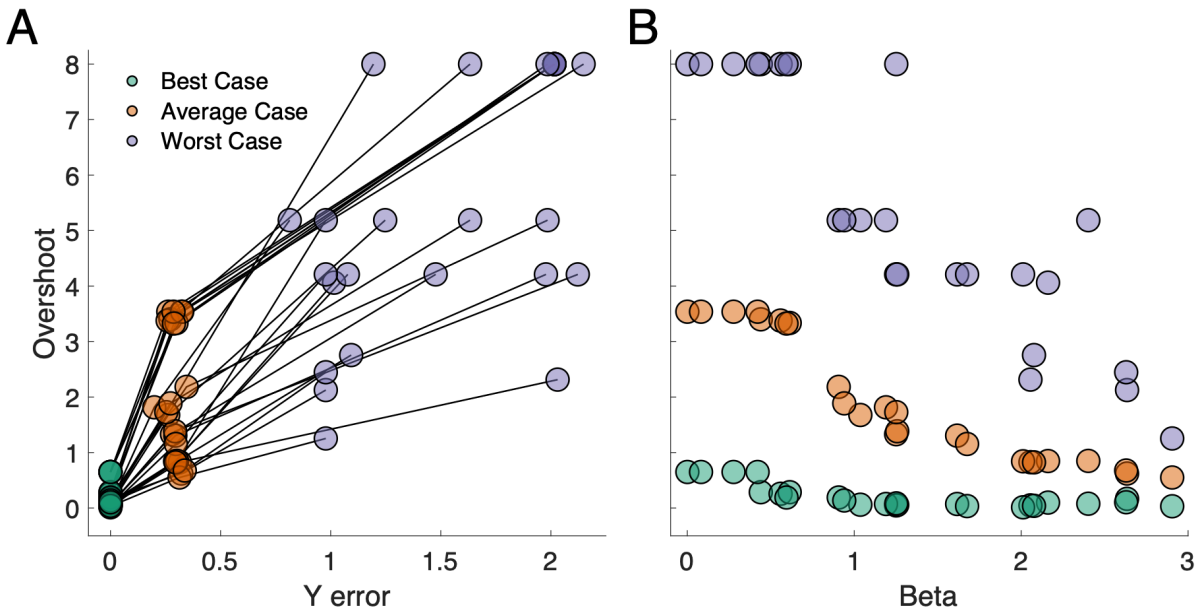

Figure 5.5: Performance of META-SFE configuration parameters *in silico*.
(A) The performance of each combination of configuration parameters in terms of final error and overshoot is represented by green, orange, and purple circles connected with black lines. (B) The overshoot of each combination of configuration parameters as a function of β.

5.3.4 In vivo validation

To validate our simulation experiments, standard Bayesian optimization was compared to the Bayesian optimization algorithm with learnable safety constraints and hyperpriors *in vivo*. For the safe-opt algorithm, the η, β, and α configuration parameters

with the lowest final error and overshoot in the *in silico* prototyping were used. For the standard Bayesian optimization algorithm, η was set to 0.4 to be consistent with previous DS optimization experiments [37], [61].

5.3.4.1 Bayesian optimization for DA maximization

In the first subject the stimulation amplitude parameter space was restricted to ±0-5 V. After the first three burn-in samples at ±0, 0.5, and 1.0 V the algorithm immediately selected the stimulation amplitude of ±4.0 V, followed by limited samples at ±5.0 V before it ultimately converged to an estimated optimal stimulation amplitude of ±3.95 V (**Figure 5.6**). In the validation phase, the DS at ±4.0 V was not significantly different than under sham conditions, and the DA at ±4.0 V was higher than at sham (DA=0, by construction; **Figure 5.7**). In a second subject, when the algorithm selected a stimulation amplitude of ±4.0 V, the stimulation induced a Racine 5 behavioral seizure with rearing and falling. The fifth stimulation amplitude sampled by the algorithm, ±1.85 V also induced a seizure. Ultimately the algorithm converged on an estimated optimal of ±0.7 V, which was validated against sham and ±1.5 V, which produced lower DA and DS, respectively.

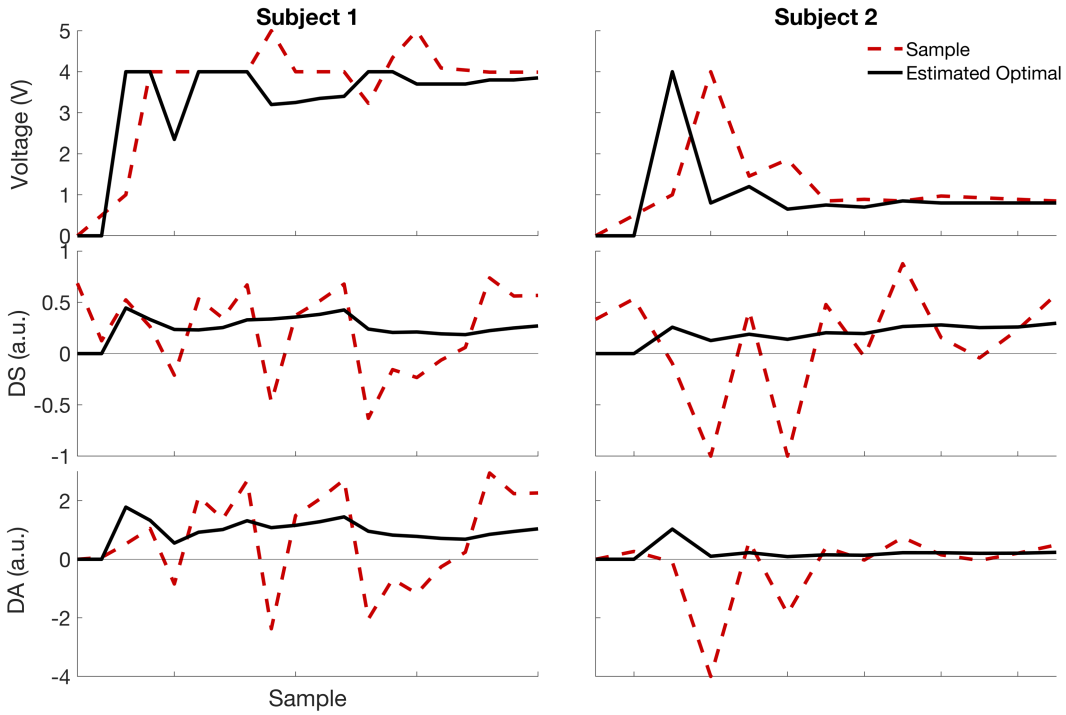

Figure 5.6: Optimization trajectory for DA maximization.

Each column shows the optimization trajectory consisting of the input – voltage, the safety function – DS, and the objective function – DA. The red dashed line indicates the stimulation samples collected from the subject, while the thick black line indicates the estimate of the optimal stimulation amplitude and resulting DS and DA according to the surrogate model. The thin gray line indicates threshold of zero on the DS and DA trajectories.

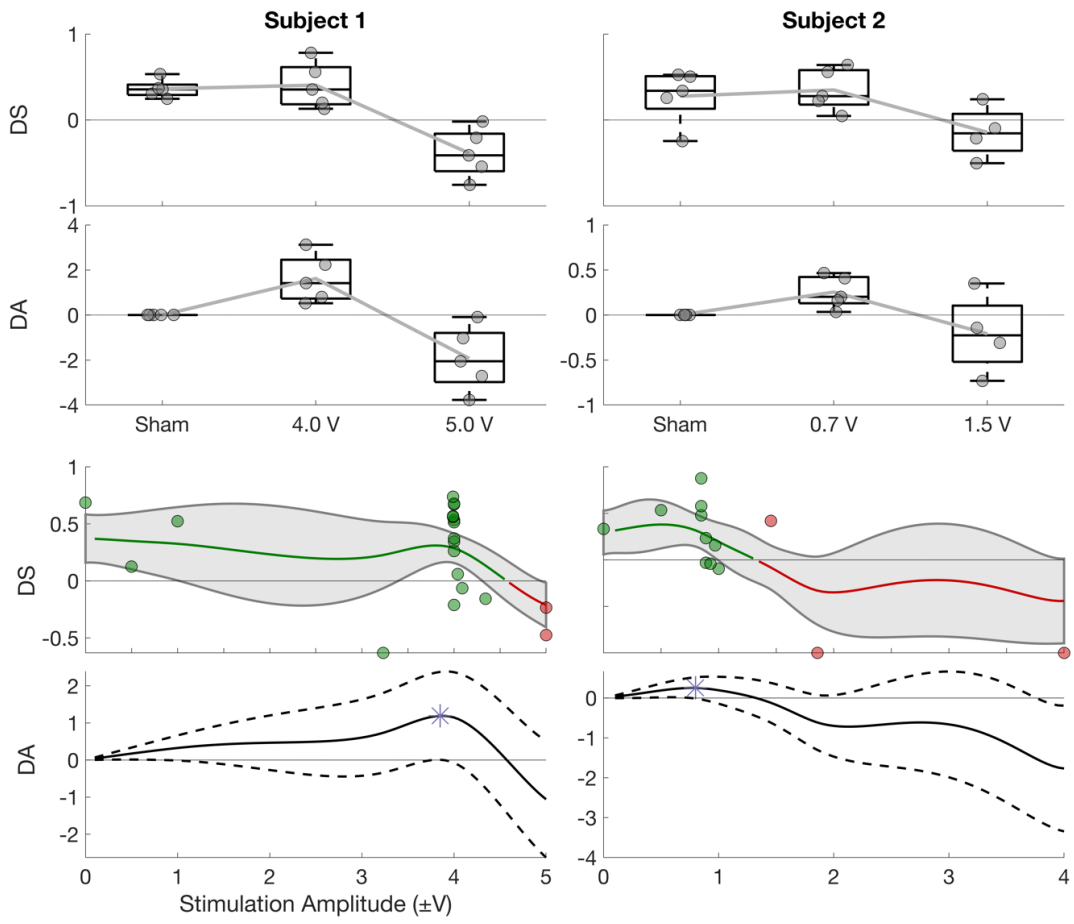

Figure 5.7: Validation of Bayesian optimization for maximizing DA.
(Top) The DS and DA under sham, optimal, and control stimulation amplitudes. Each circle represents a single trial, box plots indicate range and quartiles, gray line connects the mean from each condition. (*Third row*) GP model of the expected DS and confidence interval for a subject using all data collected over the study. The expectation of the model is shown with a green/red line indicating the safe/unsafe regions of the parameter space. Circles indicate samples collected during the optimization phase, colored green/red as inside or out of the safe region of the parameter space. Gray patch indicates the 95% confidence interval estimated from the GP. (*Bottom*) The DA objective function estimated from the model. Star indicates the optimum of the objective function.

The META-SAFE algorithm designed in Chapter 5, Section 3.3 was then deployed in *in vivo* in two additional subjects. In both subjects, the selection of stimulation amplitudes to sample was much more conservative, with ±2.1 V as the highest amplitude sampled (**Figure 5.8**). In the validation (**Figure 5.9**) for Subject 4, the control stimulation at ±2.3 V

produced a negative DS, while for subject 5, the initial control point at ±2.1 V did not cause a memory deficit. However, an additional control stimulation at ±3.15 V did cause a slight decrease in performance. These findings indicate that the META-SAFE algorithm can more carefully avoid unsafe regions of the parameter space, but may not fully explore the entirety of the safe regions.

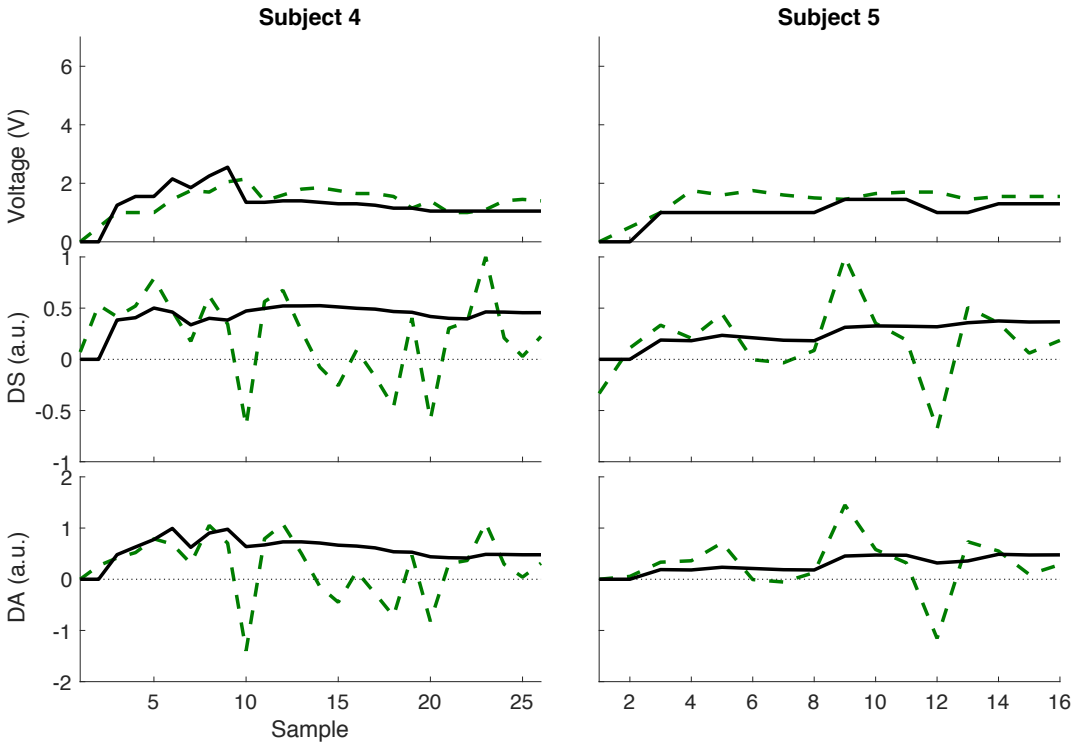

Figure 5.8: Optimization trajectory for META-SAFE algorithm.

Each column shows the optimization trajectory consisting of the input – voltage, the safety function – DS, and the objective function – DA. The green dashed line indicates the stimulation samples collected from the subject, while the thick black line indicates the estimate of the optimal stimulation amplitude and resulting DS and DA according to the surrogate model. The thin gray line indicates threshold of zero on the DS and DA trajectories

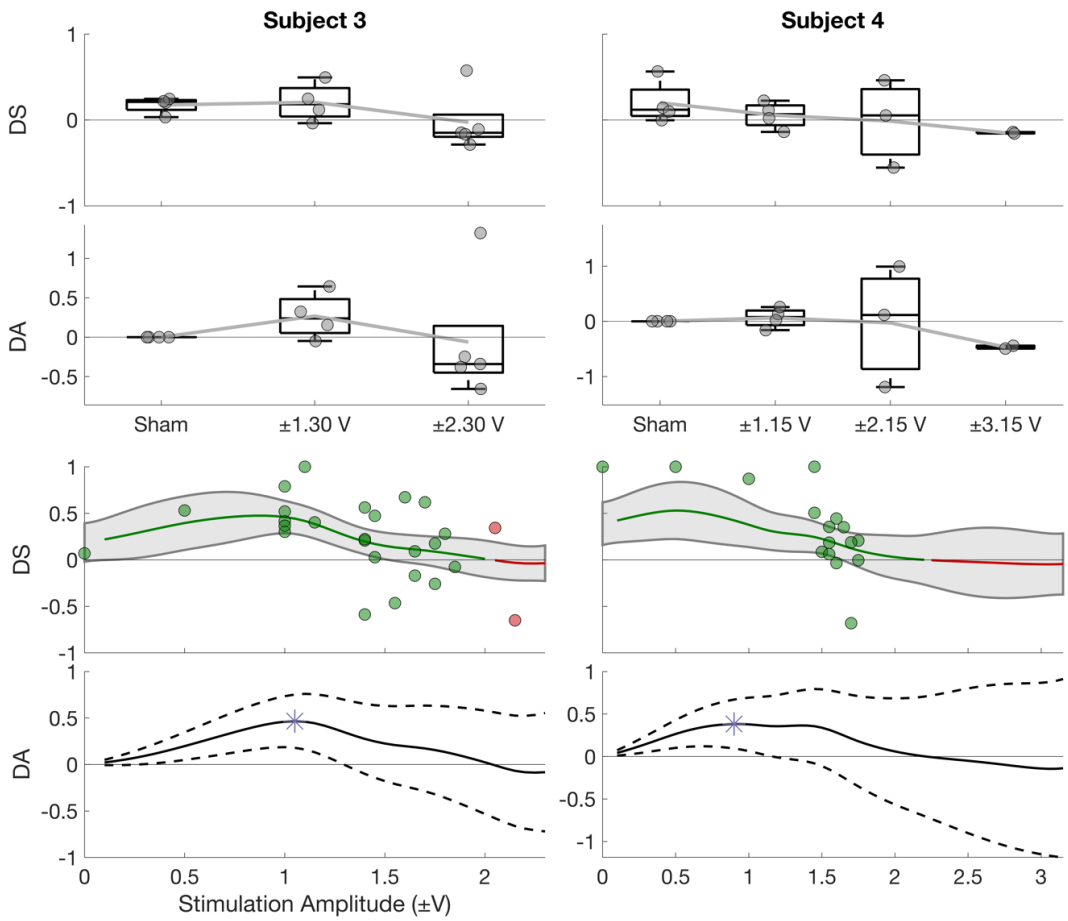

Figure 5.9: Validation of META-SAFE in vivo.

(*Top*) The DS and DA (under sham, optimal, and control stimulation amplitudes. Each circle represents a single trial, box plots indicate range and quartiles, gray line connects the mean from each condition. (*Third Row*) GP model of the expected DS and confidence interval for a subject using all data collected over the study. The expectation of the model is shown with a green/red line indicating the safe/unsafe regions of the parameter space. Circles indicate samples collected during the optimization phase, colored green/red as inside or out of the safe region of the parameter space. Gray patch indicates the 95% confidence interval estimated from the GP. (*Bottom*) The DA objective function estimated from the model. Star indicates the optimum of the objective function.

5.4 Discussion

5.4.1 Summary

In this study we designed and characterized a novel approach for safely optimizing brain stimulation. Using a previously described framework for designing and validating optimization algorithms for neural modulation we demonstrated how Bayesian optimization with learnable safety constraints can efficiently maximize a noisy objective function based on a memory task, while simultaneously avoiding regions of the parameter space that may disrupt memory.

5.4.2 Interpretation of results

The most evident challenge of this neural modulation optimization problem was the magnitude of the noise – the sample-to-sample variability – of this behavioral task. Across the models generated from the eight animals in this study, the average CNR was below 0.5, while in other studies, [35], [37], [63] the CNR of the objective function was closer 5.0. Based on this noisy objective and the extended time period to collect each sample (1/day) we restricted our analysis to Bayesian optimization based approaches. While the standard Bayesian optimization approach successfully identified optimal stimulation settings, the exploration mechanism of the algorithm led it to test the extremes of the parameter space, which ultimately resulted in side-effects for one of the subjects.

In contrast, META-SAFE approach was able to slowly broaden the search space, avoiding extremes of the parameter space unless necessary to find the optimum of the objective function. Given the noisy data, hyperpriors were also useful to prevent premature

convergence or overshoot. While we evaluated the effect of three different configuration parameters on the performance of the overall META-SAFE algorithm, only β, the parameter controlling the magnitude of the safety constraint, had a substantial effect on performance. By increasing β, we could limit by how much the stimulation amplitudes would overshoot the optimum with little effect on the other performance metric, final error.

5.4.3 Related work

Data-driven optimization approaches have been gaining traction for tuning brain stimulation in both clinical and pre-clinical studies. The foundational work applying data-driven, particularly Bayesian, optimization to the field of neuroscience was in the context of designing the optimal audio-visual stimuli for an fMRI experiment [35]. Since, Bayesian optimization has been applied *in silico* for optimizing tACS stimulation [64], tuning a linear controller system for regulating subthalamic nucleus stimulation for PD [65], and for optimizing STN stimulation for multiple, mutually competitive objectives [63]. Additionally, it has been deployed *in vivo* for modulating hippocampal gamma (32-50 Hz) power with optogenetic stimulation of the medial septum [37]. While the approaches have been found to be effective, only the multi-objective optimization algorithm accounted for side-effects, but lacked a mechanism for avoiding side-effects or unsafe regions of the parameter space. In the approach presented here, the algorithm actively learns a surrogate model for safety and uses it to avoid regions of the parameter space expected to be unsafe.

5.4.4 Future work

While we explored one possible implementation of Bayesian optimization with learnable safety constraints, safe optimization is a nascent field with ongoing advances in the algorithms [58], [66]. In particular, these safe-optimization approaches can leverage surrogate models more appropriate for high-dimensional parameter spaces [67] to accommodate more complex stimulation settings. The mechanisms for learnable safety constraints can also be incorporated into multi-objective and state-dependent optimization [48] optimization problems. Ultimately, all of these approaches will be critical for developing safe and effective automatic optimization approaches for neural modulation.

5.4.5 Limitations of the study

The results of this study should be interpreted based on the overall goal of designing a safe optimization algorithm using a proof-of-concept model system. While asynchronous distributed stimulation is a potential therapeutic option for drug-resistant temporal lobe epilepsy, and it will be crucial to consider any memory side-effects, this study was not designed to provide insight into the clinical or cognitive effects of the therapy. The algorithm designed for safely maximizing DA is specific to this particular optimization problem. However, these results show that the overall framework can be used to design problem-specific safe optimization algorithms – even for very noisy behavior-based objective functions.

5.4.6 Conclusions

Bayesian optimization with learnable safety constraints can safely and efficiently identify stimulation settings that maximize a noisy behavior-based objective function while avoiding regions of the parameter space that may be unsafe.

5.5 Supplement

5.5.1 Maximization and minimization of DS

Bayesian optimization with objective function of maximization of memory was able to sufficiently characterize the memory performance in 13 iterations for Rat 2 and 11 iterations for Rat 1 and Rat 3 (**Figure 5.10**). Bayesian optimization with objective function of minimization of memory was able to sufficiently characterize the memory performance in 14 iterations for Rat 1, 11 iterations for Rat 2, and 12 iterations for Rat 3 (Figure 9). There were no specified convergence criteria, as heretofore insufficient data had been collected. Visualizing the approximation of the objective function learned by the Bayesian optimization algorithm, it can be shown that each rat responded differently to the stimulation. Rat 1 showed memory impairments from any stimulation over 0 V. Rat 2 was able to tolerate voltages up to 1 V before showing memory impairments. Rat 3 was able to tolerate voltages up to 1.5 V before showing considerable memory impairments. **Figure 5.11** makes it clear that increasing voltage stimulation did not result into an aversive stimulus that causes the rats to prefer stationary object.

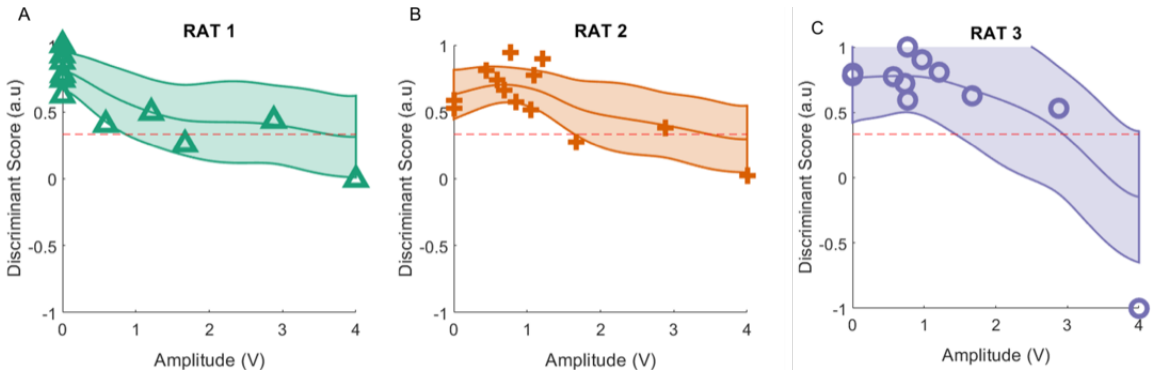

Figure 5.10: The BaO algorithm with the objective function of maximization of memory.

(A-C) Memory performance with increasing voltage stimulation for Rats 1-3, respectively. Stimulation frequency was set at 7Hz. Red dotted line indicates a threshold score, performance above which demonstrates memory. Spatial object recognition task was used to test memory in normal rats during active ADMES stimulation. The shaded area indicates the confidence interval of the GP model.

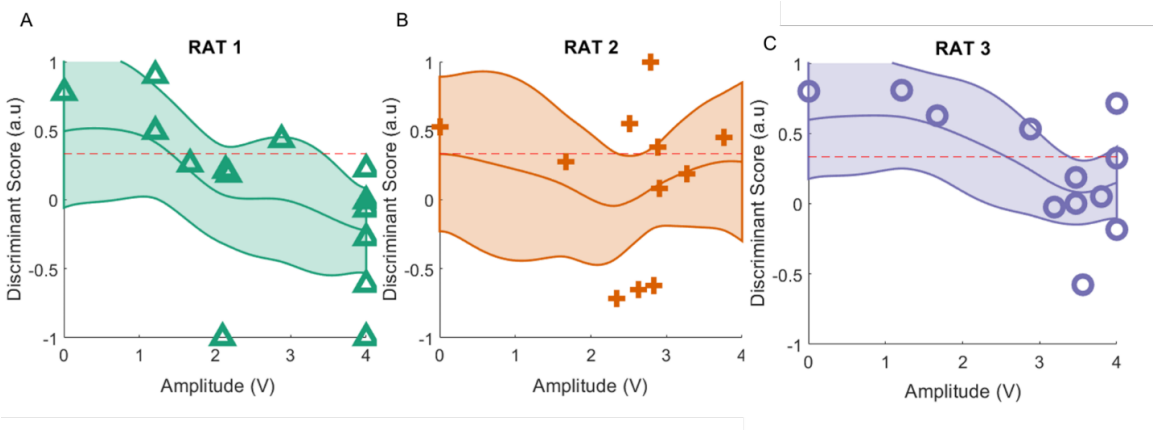

Figure 5.11: The BaO algorithm with the objective function of minimization of memory.

(A-C) Memory performance with increasing voltage stimulation for Rats 1-3, respectively. Stimulation frequency was set at 7Hz. Red dotted line indicates a threshold score, performance above which demonstrates memory. Spatial object recognition task was used to test memory in normal rats during active ADMES stimulation. The shaded area indicates the confidence interval of the GP model.

CHAPTER 6. APPLICATION OF THE FRAMEWORK FOR MULTIPLE OBJECTIVES

6.1 Introduction

6.1.1 DBS for PD

As described in Chapter 1, Section 1.1, a common example of DBS is electrical stimulation of the subthalamic nucleus (STN) for the treatment of motor symptoms (e.g. tremor, bradykinesia) in PD [1], [16], [68], [69]. A major challenge for the therapy is finding a setting that stimulates the STN region to maximally reduce motor symptoms, while avoiding the corticospinal/bulbar tract in the internal capsule – a neighboring white matter pathway – which can induce involuntary muscle contractions, speech problems, or pain [55], [70]. The current standard of care is to apply different stimulation settings in clinic and observe the patient's clinical signs and feedback regarding side effects, which can be subjective and time-consuming [12], [13], [70]. To address this problem, there is active research into electrophysiological and kinematic biomarkers that may be used to rapidly and quantitatively evaluate the effectiveness of a stimulation setting [17], [27], [71]–[75]. Cortical (cEP) and motor (mEP) evoked potentials have been investigated as putative biomarkers for stimulation effectiveness (activation of cortico-subthalamic hyperdirect pathway) and motor side effects (activation of corticospinal/bulbar tract), respectively [76]–[79]. While the cEP and mEP measures have not yet been prospectively validated as biomarkers for STN DBS programming, they capture the inherent trade-off

between symptom control and side effects, and can be used as a model system for developing novel approaches for DBS programming.

6.1.2 Multi-objective optimization for DBS programming in PD

One approach is to frame DBS programming as a multi-objective optimization problem, for which many potential engineering solutions exist [80], [81]. From this perspective, the goal of DBS programming is to simultaneously address multiple, often mutually competitive, objectives – controlling symptoms while minimizing side effects. The optimal trade-off between the objectives is referred to as the Pareto set (**Figure 6.1**). In the context of DBS programming, the Pareto set would be the subset of stimulation settings that are superior to all others in at least one of the symptom or side effect objectives. In contrast, the stimulation settings not in the Pareto set are those that are inferior to at least one other stimulation setting on all objectives. The goal of a multi-objective optimization algorithm in this study is therefore to accurately identify which stimulation settings are in the Pareto set – a substantially smaller subset of the overall parameter space. This could reduce the size of the parameter space that needs to be evaluated clinically, and in turn may accelerate DBS programming and allow for more effective stimulation settings to be identified. Rather than viewing this optimization approach as a method to determine one optimal setting, calculating the Pareto set allows flexibility to consider competing demands (e.g., changing the level of acceptable side effects depending on clinical response and patient preferences).

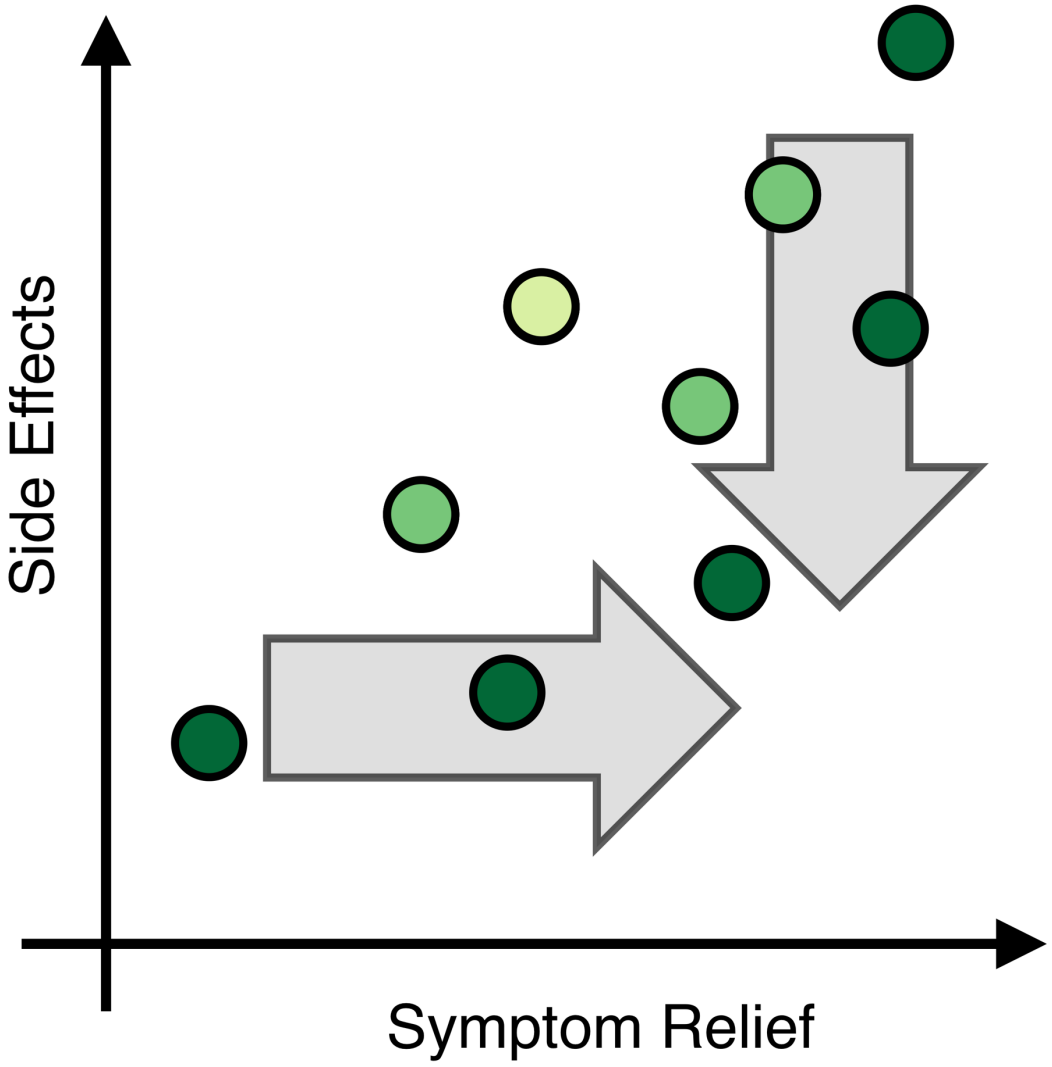

Figure 6.1: Pareto set as the optimal trade-off between symptom relief and side-effects.

Each circle represents the effect of a single stimulation setting on both symptom relief and side effects. The Pareto set are those stimulation settings farthest to the right (maximizing symptom relief) and bottom (minimizing side-effects) shown in dark green. In addition to the optimal trade-off, all other stimulation settings can be ranked in terms of favorability with the Pareto set having a rank of 0. In this example, the light green points have a worse ranking (1) than the Pareto set (0), but a better ranking than the yellow-green point (2).

6.1.3 *Extension of framework for multiple objectives*

In this study, we extended a previously described framework for designing data-driven optimization systems for neural modulation [37]. Originally developed for single objectives, we adapted this framework to the multi-objective optimization of the cEP and mEP neural features (**Figure 6.2**). Following this framework, we used intraoperative cEP and mEP data to construct a set of statistical ground-truth models that captured the response of these features to STN stimulation for each patient. We then performed simulation experiments in which we sampled from these models for high-throughput prototyping of different multi-objective data-driven algorithms, to identify the algorithmic components necessary for successful optimization. While the resultant design information is specific to the putative cEP and mEP neural features, the overall framework can be readily adapted to ongoing research and newly discovered biomarkers [17], [27], [75], [82], [83].

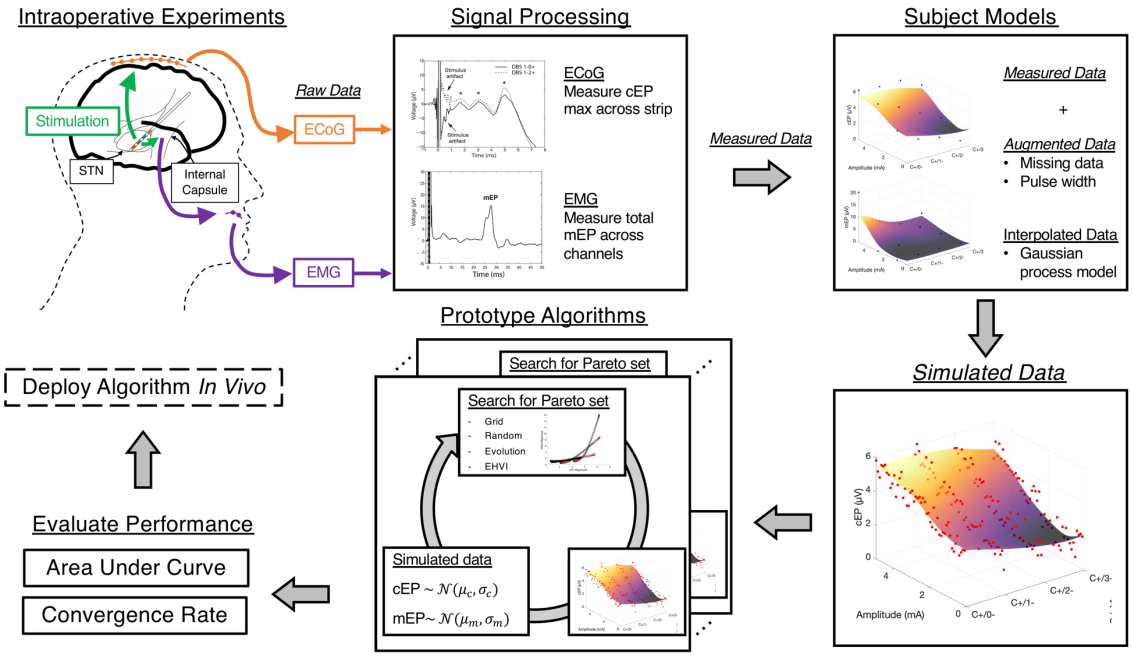

Figure 6.2: Framework for designing multi-objective optimization algorithms for STN DBS.

Clockwise from top left. During the procedure to implant the DBS device, the clinical electrode is used to stimulate the STN while ECoG and EMG are being recorded as *raw data*. The *raw data* is filtered and the cEP and mEP electrophysiological features are annotated. These features, the *measured data*, are then combined with *augmented data* to fill in missing data points and approximate higher dimensional parameter spaces (e.g. the effect of pulse-width). The *measured data* and the *augmented data* are then fit to a Gaussian process model to represent the input -> output (stimulation parameters -> cEP/mEP) relationships derived from an individual subject. These models allow for *interpolated data* to be estimated for stimulation settings that were not applied in the intraoperative experiments. These models are then used for high-throughput prototyping of different algorithms on the task of identifying the optimal trade-off between the two objectives – maximizing cEP and minimizing mEP – known as the Pareto set. Since Gaussian processes naturally estimate the mean and variance of the output, they can generate noisy *simulated data* that can be used to evaluate the performance of data-driven optimization algorithms in a realistic context. Each algorithm is evaluated in terms of its accuracy (AUC) and rate at which it identifies the Pareto set. Finally, the algorithm that performs best will be well-poised for deployment *in vivo* in future studies.

6.2 Methods

The organization of the methods follows the two key steps in the design framework: 1) data collection and modeling, and 2) high-throughput prototyping of different algorithms. Section 6.2.1 describes how the dataset used in this study was collected, how it was processed, and how our cEP and mEP neural features were obtained. Then, section 6.2.2 uses this data to construct ground-truth statistical models for each subject for the purpose of testing a range of optimization approaches *in silico*. Section 6.2.3 formally defines the solution to a multi-objective optimization problem, the Pareto set, and how it was calculated. Section 6.2.4 details a set of optimization algorithms chosen to highlight particular characteristics of multi-objective optimization approaches. Finally, section 6.2.5 defines performance metrics specific to this clinical context that were used to compare the different algorithms.

6.2.1 Data collection

6.2.1.1 Electrode implantation and biomarker recordings

Subjects included in this study were undergoing surgery to implant electrodes in the STN as part of the standard of care for treatment of medically refractory PD. As part of an additional research study, a 28-contact subdural ECoG recording strip (Ad-Tech) was temporarily placed over the motor cortex ipsilateral to the stimulating electrode. Activity of up to 8 contralateral muscles were recorded with surface EMG electrodes. The relevant experimental methods are summarized below, but additional details describing the protocol can be found in a prior publication where data from patients 1 and 2 were presented [76]. This study was approved by the Emory University Institutional Review Board.

6.2.1.2 Grid search experiments

The *raw data* were collected by applying stimulation via the clinical lead in the STN while simultaneously recording from the ECoG and EMG electrodes. A parameter sweep was performed where a pre-specified set of stimulation settings was applied. Each stimulation setting consisted of a different combination of current amplitude, active (negative or cathode) contact, and return (positive or anode) contact. The order of the stimulation settings was randomized, and sequentially applied for 12 seconds followed by a 3 second pause before the next stimulation. Stimulation pulse width was set to 60 μs and stimulation frequency to 10 Hz. Stimulation could be monopolar – where a contact on the electrode is used as the cathode and a shoulder patch is used as a remote return/anode, or bipolar – where one contact of the electrode is the cathode and another is the anode. The

DBS lead model was Medtronic 3389 for subjects 1-4 and Abbott 6172 for Subject 5. All subjects had each monopolar configuration tested at 1, 3, and 5 mA (pseudo-ring settings were used for the segmented electrode in subject 5). Subsets of bipolar configurations were tested at the same amplitudes, although some combinations of configurations and amplitudes were not tested in some subjects. Specific data used are detailed in **Table 6.1**.

6.2.1.3 Measuring cortical and motor evoked potentials

ECoG and EMG signals were recorded at a 22 kHz sampling rate using a Neuro Omega data acquisition system (Alpha Omega Engineering). ECoG was re-referenced with a bipolar montage between adjacent contacts along the recording strip to create 26 bipolar recording channels. Stimulation pulse onset was automatically detected from the ECoG or EMG channel with the largest stimulation artifact. Segments were time-aligned on pulse onset, averaged to resolve the evoked potential, and peak/trough amplitudes and latencies were visually determined. The amplitude of the evoked potential between 2-4 ms was measured for each bipolar ECoG channel as the cEP, indicating the degree of antidromic cortico-subthalamic hyperdirect pathway activation. The maximum amplitude of cEP across ECoG channels was used as the putative biomarker of therapeutic benefit.

The mEPs were manually annotated as the signal peak for each EMG channel. Signals where no clear peak was visible at the expected latency were annotated as 0 μV. The sum amplitude of the mEP across the available EMG channels was used as a biomarker of a motor side effect.

6.2.1.4 Data augmentation

For some subjects, data were not collected for stimulation settings on the borders of the bipolar parameter space (3+/0- and 0+/3-). As the statistical models used to represent the data, described in Chapter 6, Section 2.2, are better suited to interpolation rather than extrapolation, the initial ground-truth model estimates in these regions of the parameter space were not physiologically appropriate. To compensate for this, artificial data points were manually added to the ground-truth models in the form of *augmented data*. (**Table 6.1**). These data points had inputs corresponding to the missing stimulation setting and outputs were proportional to those of the patients with complete data, where the wide bipolar setting has a similar, but smaller effect than for the monopolar setting at the same contact and amplitude.

While there were not enough data to directly model the effect of different pulse widths on the two objectives, limited experiments applying stimulation at 20, 60, and 120 μs indicated a sigmoidal increase in cEP with increasing pulse width (**Figure 7C** in [84]). To incorporate pulse width, the bipolar ground-truth models were augmented with data created by passing known cEP and mEP values through a sigmoid function. These data augmentation steps may have produced ground-truth models that were less consistent with the original specific subjects. However, the intention was to capture the appropriate characteristics and diversity to suitably measure the performance of potential optimization algorithms, particularly in higher-dimensional parameter spaces.

6.2.2 Constructing ground-truth models for simulation

The data from the intraoperative experiments were then used to construct simulation models for each subject. The goal of these models was to allow an optimization algorithm to interact with a black-box function that, for a given set of stimulation parameters, would produce a stochastic response consistent with previous experiments. These models represented 'known' ground-truths corresponding to individual subjects that provided realistic conditions to prototype and evaluate a data-driven optimization algorithm.

6.2.2.1 Spatial organization of stimulation parameters

The first step to construct the ground-truth models was to determine the organization of the parameter space (**Figure 6.3**). Monopolar stimulation was modeled as a two-dimensional parameter space consisting of the four cathodes and 50 pulse amplitudes ranging from 0.1 mA to 5.0 mA with an increment of 0.1 mA, similar to the increments used in DBS programming [12]. All combinations of cathode and amplitude created a parameter space with a total of 200 potential stimulation settings (**Figure 6.3a**). This monopolar (2D) parameter space produced two surfaces for the cEP and mEP biomarkers (**Figure 6.3b,c**). Bipolar stimulation was modeled as a three-dimensional parameter space consisting of 16 cathode/anode pairs and 50 pulse amplitudes, for a total of 800 potential stimulation settings (**Figure 6.3e**). Monopolar configurations were also included in the bipolar model and assigned to the coordinates where cathode and anode are the same (diagonal across parameter space). To visualize the objective function on a bipolar (3D)

parameter space, surfaces were constructed for multiple amplitude cross-sections (**Figure 6.3f**). A four-dimensional parameter space – bipolar/pulse width (4D) – was modeled using the bipolar stimulation model with an additional 6 pulse widths (20 μs to 120 μs with 20 μs increments), for a total of 4,800 potential stimulation settings. The bipolar/pulse width models were visualized using multiple bipolar models, each a three dimensional cross-section for a fixed pulse width (**Figure 6.3g**).

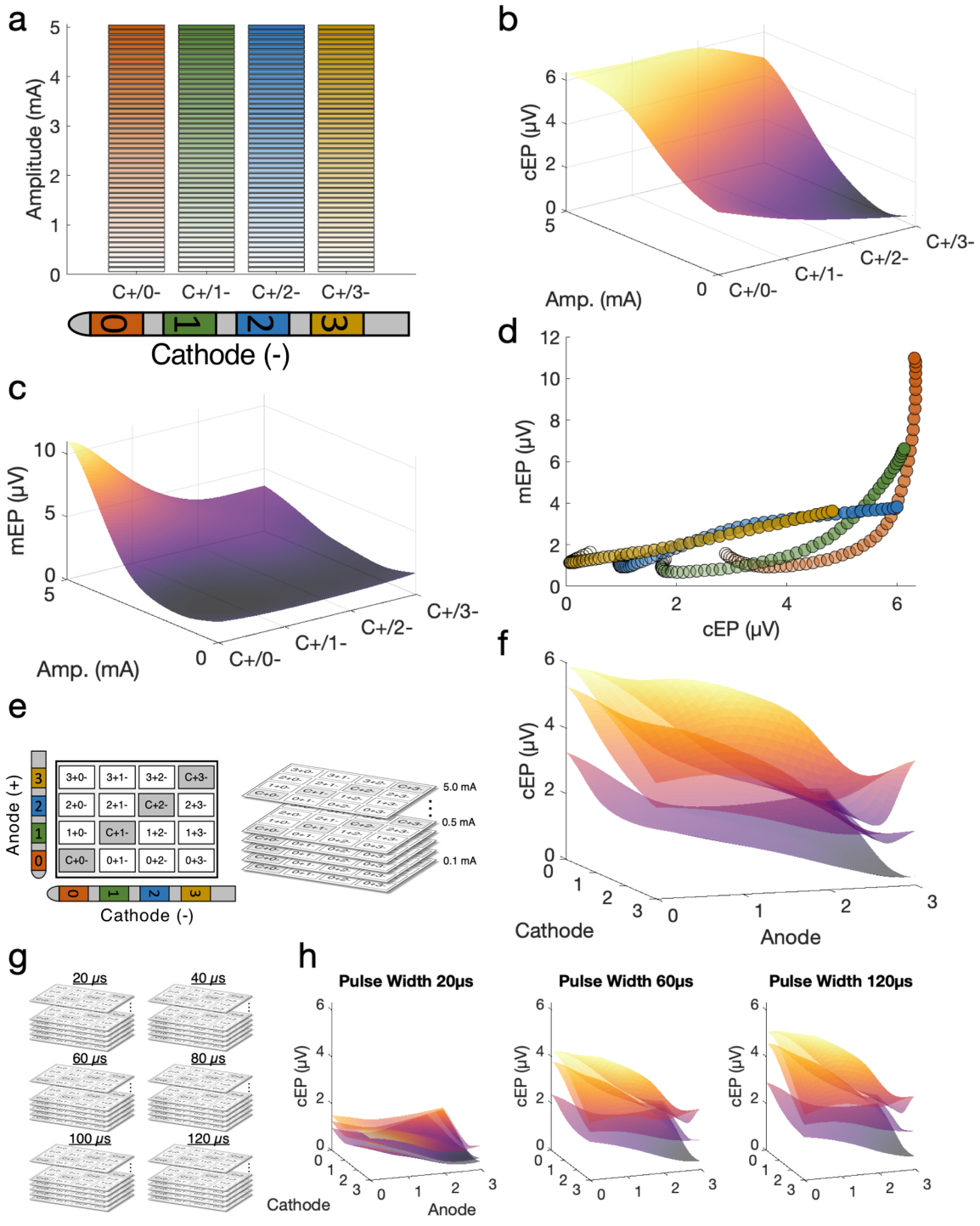

Figure 6.3: Parameter and objective space representation of STN DBS data.

(a) Spatial arrangement of the monopolar (2D) model. (b and c) Ground truth models for cEP and mEP, respectively, for Subject 1. Surface is the expectation of the Gaussian process model. Color gradient is for visualization. (d) The objective space for the ground truth model. Colors correspond to the monopolar cathode used for stimulation (as shown in 2a), estimated at 50 different amplitudes indicated by the transparency of the circle, where 5 mA is fully opaque. (e) Spatial arrangement of the bipolar (3D) model. (f) Ground-truth model of bipolar stimulation. The surfaces represent a cross section of the model at 1, 3, and 5 mA. (g) Spatial arrangement of the bipolar/pulse width (4D) model. (h) Bipolar cross-sections of a ground-truth model at 20, 60, and 120 μs.

6.2.2.2 Model fitting

Each ground-truth model consisted of two Gaussian processes (GP) models that estimated cEP and mEP outputs for a given stimulation setting [45]. A Gaussian process is composed of a mean function:

$$\mu = m(\mathbf{x}') \tag{6.1}$$

and a covariance function:

$$\Sigma = k(\mathbf{x}, \mathbf{x}') \tag{6.2}$$

where $\mathbf{x}$ is a vector containing the data used to construct the model, and $\mathbf{x}'$ is the sample point. In our application, the inputs to the model were restricted to those detailed in Chapter 2 Section 2.2. The output was the cEP and mEP drawn from Gaussian distributions:

$$cEP, mEP \sim \mathcal{N}(\mu, \Sigma) \tag{6.3}$$

In our implementation, we used a GP with a constant function based on the mean of the data, and a third order Matérn kernel with automatic relevance determination for the covariance:

$$k(x_i, x_j) = \sigma_f^2\left(1 + \sqrt{3}r\right)\exp\left(-\sqrt{3}r\right) \tag{6.4}$$

where σ_f^2 is the variance of observation and:

$$r = \sqrt{\sum_{m=1}^{d} \frac{\left(x_{im} - x_{jm}\right)^2}{\sigma_m^2}} \tag{6.5}$$

where σ_m^2 is the set of length scale hyperparameters which describe the variance of x_m, [45]. In other words, σ_m^2 represents how much the expectation can vary only a given dimension of the parameter space. Each model was individually reviewed and model hyperparameters (mean, length scale, signal noise, measurement noise) were manually adjusted in the event of overfitting or non-physiological responses.

6.2.2.3 Simulation Model

After fitting the ground-truth cEP and mEP models, they were wrapped in a simulation function to replicate the noise or stochasticity inherent to any biological system. For a feasible stimulation setting defined in 6.2.2.2., the simulation function calculates the mean (μ) and standard deviation (σ) from the ground-truth GP models. The model outputs

are drawn from a normal distribution with μ and σ to output an appropriately stochastic response to stimulation .

6.2.3 Estimating the Pareto Rank

The goal of multi-objective optimization is to identify the Pareto set. In the context of DBS programming, the Pareto set is defined as set of stimulation settings that are non-dominated. A stimulation setting is non-dominated if there is no other stimulation setting that can improve on one objective (e.g. increasing cEP) without worsening on another objective (e.g. increasing mEP) (**Figure 6.1**).

Mathematically [85], for a system $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$, where X is a set of feasible inputs on $\mathbb{R}^n$, and Y is the feasible set of objectives in $\mathbb{R}^m$, such that:

$$Y = \{y \in \mathbb{R}^m : y = f(x), x \in X\} \tag{6.6}$$

A point $y''^{\in \mathbb{R}^m}$ strictly dominates another point $y'^{\in \mathbb{R}^m}$, if y'' is superior to y' on all objectives, written as $y'' \succ y'$. Therefore, the Pareto set $P(Y)$ is defined as:

$$P(Y) = \{y' \in Y : \{y'' \in Y : y'' \succ y', y' \neq y''\} = \emptyset\} \tag{6.7}$$

While inclusion in the Pareto set is binary, a relative Pareto rank can be calculated recursively [86]. The recursive dominance order ranks stimulation settings by first calculating the Pareto set, assigning it a rank of 0, then removing it. The Pareto set is then recalculated, assigned a rank of 1, and the process is repeated until no more stimulation

settings are left. Therefore, a stimulation setting with rank 0 would be in the Pareto set, and settings with better ranks (lower numbers) would be close to the Pareto set and nearly optimal, while settings with worse ranks (higher numbers) would be less optimal. The Pareto rankings were then used to label all the stimulation settings in the parameter space on a gradient where a cut-off could be specified for what is considered clinically relevant (**Figure 6.4**). Increasing the cut-off would cause a larger parameter space to be labeled as a member of the Pareto set (less specific), but more likely to include the true optimal for a given subject (more sensitive). We specified this cut-off based on the minimum ranking necessary to include the stimulation found to be optimal through traditional DBS programming.

The Pareto ranking for the ground-truth cEP and mEP models was calculated across each potential stimulation setting to create a known solution to the multi-objective optimization problem. Therefore, the goal of the optimization algorithm was to correctly estimate the Pareto ranking for all stimulation settings while testing as few settings as possible. We evaluated three different approaches for estimating the Pareto ranking as additional data was collected during the optimization.

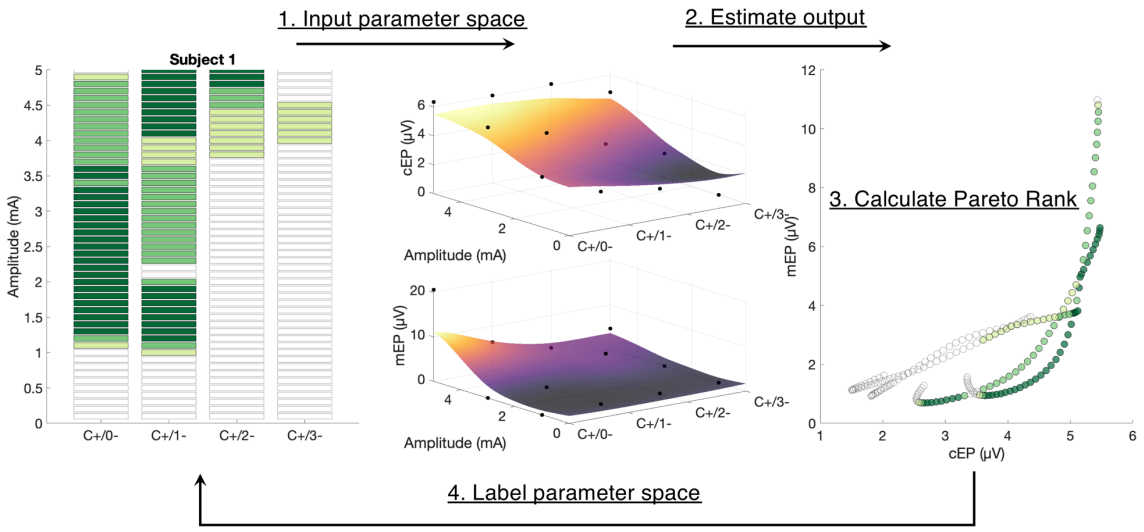

Figure 6.4: Mapping from parameter space to objective space to estimate Pareto rank.

Left, The set of feasible stimulation settings in the monopolar parameter space. The top 3 Pareto ranks are highlighted as Pareto set (dark green), rank 1 (green), and rank 2 (light green). Stimulation settings with a Pareto rank greater than 2 are open grey boxes. *Middle*, The cEP and mEP ground-truth models for Subject 1. Surface is the expected output for a given stimulation setting. Data points used to fit model are black dots. *Right*, The objective space. Each circle corresponds to a specific stimulation setting and is color coded by Pareto rank as in the parameter space.

6.2.3.1 Pareto set based on measured data

In the simplest approach, the optimization algorithm calculated the Pareto set using the actual noisy measured data. Within the parameter space defined in Chapter 6, Section 2.2, the Pareto ranking was calculated for stimulation settings that were actually tested. All other untested stimulation settings were then assigned to one Pareto rank worse/higher than the worst/highest among the tested stimulation settings.

6.2.3.2 Pareto set based on surrogate model data

In a second approach, the optimization algorithm estimated the effect of all the stimulation settings, tested and untested, using a surrogate model. A surrogate model is a typically statistical model that is fit to the limited stimulation/measurement data collected during the optimization process. Two GPs, as described in section Chapter 6, Section 2.2.1, were used as surrogate models to estimate how each stimulation setting in the parameter space would affect the cEP and mEP. While the surrogate models used the same statistical structure as the ground-truth models, they were only fit to the noisy data collected during the optimization process, and had no prior information on the ground-truth Pareto ranking. After each additional stimulation setting wis tested on the ground-truth models, the data were added to the models, the hyperparameters were re-fit, the entire parameter space was re-estimated, and each stimulation setting is assigned an estimated Pareto rank.

6.2.3.3 Hyperpriors

In a third approach for estimating the Pareto set, information from other subjects – in the form of hyperpriors – was incorporated when fitting the surrogate models, which could improve the estimate of the parameter space and the accuracy of the Pareto ranking. Hyperpriors are used to specify a prior distribution for the hyperparameters in a GP model. For example, if a stimulation setting has a relatively small effect in four subjects, but a very large effect when first tested in a fifth subject, the hyperprior will help attribute that data point to noise, rather than reliable data [41]. This can be especially useful when little data has been collected, such as during the beginning of an optimization experiment. In this

study, hyperpriors were not used when constructing the ground-truth models, but were evaluated as a potential component of the optimization algorithms (Chapter 6, Section 2.4)

The hyperpriors for a given subject were created using the model hyperparameters from all other subjects. The Gaussian process kernel used in this study has hyperparameters for the mean function, the length scales, the signal noise, and the measurement noise that are selected during model fitting and are going to be specific to each subject model. To the construct a hyperprior for a given subject, the hyperparameter values were taken from all other models and the mean and standard deviation were calculated for each one, representing a prior Gaussian distribution of those hyperparameters. When fitting the model during optimization, the Gaussian distribution hyperprior was defined for each of the hyperparameters using these statistics. After each additional stimulation setting was tested on the ground-truth models, the model was re-fit using the hyperpriors, the entire parameter space re-estimated, and Pareto ranks assigned to each possible stimulation setting.

6.2.4 Multi-objective surrogate-model optimization

There is no established approach for efficiently sampling stimulation settings for programming STN DBS or other neural modulation applications, although some protocols and heuristics have been described [12], [13]. As such, we sought to broadly compare sampling strategies that represented different classes of algorithms. Following the framework, ground-truth models wrapped in the simulation function were used to prototype and evaluate four different strategies for sampling the parameter space: repeated

samples from a fixed grid, a simple surrogate-model evolutionary approach, random sampling, and Bayesian optimization with an expected hypervolume improvement (EHVI) acquisition function.

6.2.4.1 Fixed Grid

Ten stimulation settings were randomly selected from the parameter space based on a uniform distribution without replacement. Each stimulation setting was sampled 10 times in a random order. While 10 samples of the same stimulation setting may not be necessary in practice, this allowed for an equitable comparison to the other search strategies with 100 samples. After each sample, the Pareto ranks were assigned using the three approaches defined in Chapter 6, Sections 2.3.1-2.3.3: 1) the noisy simulated cEP and mEP data, 2) the entire input space estimated using surrogate models fit to the simulated data, and 3) fitting the GP models to the simulated data using hyperpriors derived from all *other* subject models.

6.2.4.2 Evolutionary search

In the initial iteration, 10 stimulation settings were randomly selected from the parameter space based on a uniform distribution and then applied to the ground-truth model. The simulated data were fit to surrogate GP models and used to estimate the entire parameter space. The 10 stimulation settings with the best/lowest Pareto ranking were then sampled, and Pareto rankings iteratively recalculated. This process was repeated for 10 iterations or a total of 100 samples. Separate experiments were performed with and without using hyperpriors to fit the surrogate GP models.

6.2.4.3 Random sampling

One hundred inputs were randomly and uniformly selected with replacement from the parameter space. The Pareto set was then calculated using the three approaches described in *Fixed Grid*.

6.2.4.4 Bayesian optimization with expected hypervolume improvement acquisition function

As part of an initial burn-in period, 10 stimulation settings were randomly selected from the parameter space based on a uniform distribution and applied to the ground-truth model. The burn-in samples were then fit to surrogate GP models for cEP and mEP. The EHVI acquisition function was then used to select the sample point. This was repeated for a total of 100 samples. Separate experiments were performed as in the evolutionary search.

The goal of the EHVI acquisition function is to identify the stimulation setting expected to provide the largest increase in the hypervolume created by the estimated Pareto set [87]. For a candidate stimulation setting, the expectation μ and uncertainty σ were estimated from the cEP and mEP surrogate models constructed from the simulated data. Two new models, cEP* and mEP*, were then created by adding an additional datapoint consisting of the candidate stimulation setting, and the "best-case" outputs defined as $\mu_{cEP} + \kappa \cdot \sigma_{cEP}$ and $\mu_{mEP} + \kappa \cdot \sigma_{mEP}$, where κ is a configurable parameter within the algorithm. The Pareto set was then estimated based on the cEP* and mEP* models and the hypervolume was calculated using the reference point (0, max(mEP)). The candidate

stimulation setting that predicted the largest EHVI was found using a standard single-objective Bayesian optimization approach [44].

6.2.5 Performance metrics

After prototyping the different data-driven multi-objective optimization algorithms on the ground-truth models, the performance of the algorithms was quantified and compared.

6.2.5.1 Pareto AUC

The performance of the optimization algorithm was computed using the area under the receiver operating characteristic curve (AUC). To do this, we framed the comparison between the Pareto set estimated by the optimization algorithm and the ground-truth reference Pareto set as a classification problem. To adjust the sensitivity vs. specificity, the Pareto ranking threshold (Chapter 6, Section 2.3) for inclusion in the Pareto set could be relaxed. For example, if the threshold was set to only compare those stimulation settings with Pareto ranking equal to zero, the classifier would have perfect specificity, but no sensitivity. In contrast, if the threshold was set to include all stimulation settings, the classifier would have perfect sensitivity, but no specificity. By sweeping the Pareto rank threshold from highest specificity to highest sensitivity, the receiver operating characteristic curve was calculated to produce the AUC. The AUC was used as the primary metric to evaluate the performance of each algorithm.

6.2.5.2 Convergence rate

To measure the convergence rate, the AUC was calculated after each sample was collected. The AUC was then fit to an exponential loss model as a function of sample,

$$A * e^{-bt} \tag{6.8}$$

The coefficient b was then used as the measure for convergence rate. More negative indicated faster progression towards a high AUC

6.2.6 Analysis and statistics

A single 5-way analysis of variance (ANOVA) model with post-hoc comparison using Tukey's procedure was constructed using the final AUC and convergence across search strategy, modeling approach, dimensionality of the parameter space, subject, and trial. The relevant aspects of this ANOVA model are described in Chapter 6, Sections 3.3.1 and 3.3.2 and the full list of p-values for each predictor and interaction is shown in **Table 6.2**.

In Chapter 6, 3.3.3, linear regression is used to determine whether varying the configuration parameters of the EHVI+Model and EHVI+Hyperprior significantly affect final AUC and convergence. Significance for other comparisons was determined using a t-test.

P-values are reported; however, given the high-throughput nature of these simulation experiments, large numbers of repeated simulation trials can lead to arbitrarily low p-values.

Unless otherwise noted, simulation experiments were performed for each subject using 30 independent optimization trials consisting of 100 samples. All simulations and analysis were implemented using Matlab 2020a (Mathworks, Natick, MA) and performed on a desktop computer with an Intel i7 8th generation 8-core 3.60 GHz processor and 64 GB of DDR3 RAM.

6.3 Results

Electrophysiology data (cEP and mEP amplitudes at different stimulation settings) from 5 patients with PD were used in this analysis. The data set included complete monopolar parameter sweeps (4 contacts, 3 amplitudes each) for all subjects, and a sufficiently complete bipolar parameter sweep (10-12 contact configurations, 0-3 amplitudes each) for 4 of the subjects. The bipolar data for subjects 1 and 4 were augmented with 2 and 3 additional data points, respectively (**Table 6.1**) and all data points were replicated for each subject at pulse widths 20 μs and 120 μs to create the bipolar/pulse width (4D) models.

Table 6.1: Subject characteristics and collected data.

Subject	Age/Sex	DBS lead model	Cortical strip side/ laterality at M1, (mm)	# Muscles for EMG	Monopolar settings	Bipolar settings	Added settings (augmented dataset)
1	62/M	Medtronic 3389	R/34.0	2 face, 4 arm, 1 leg	1,3,5 mA at 0-,1-,2-,3-	1,3,5 mA at 0+1- 1+0-,1+2- 2+1-,2+3- 3+2-	3 mA at 0+3-, 3+0- All at 20 μs and 120μs
2	65/M	Medtronic 3389	R/44.0	3 arm, 1 leg	1,3,5 mA at 0-,1-,2-,3-	1,3,5 mA at 0+1- 1+0-,1+2- 2+1-,2+3- 3+2- 3 mA at 0+3-,3+0-	All at 20 μs and 120μs
3	53/M	Medtronic 3389	R/40.9	1 face, 3 arm	1,3,5 mA at 0-,1-,2-,3-	1,3,5 mA at 0+1-,0+3- 1+0-,1+2- 2+1-,2+3- 3+0-,3+2-	All at 20 μs and 120μs
4	57/M	Medtronic 3389	R/24.3	2 face, 4 arm, 1 leg	1,3,5 mA at 0-,1-,2-,3-	1,3,5 mA at 0+1-, 1+0-,1+2- 2+1-,2+3- 3+0-,3+2-	1,3,5 mA at 0+3- All at 20 μs and 120μs
5	44/M	Abbott 6172	R/18.9	2 face, 4 arm	1,3,5 mA at 0-,1-,2-,3-		

6.3.1 Clinical interpretation of the Pareto ranking

We first used the electrophysiology data to build ground-truth models for individual subjects as described in Chapter 6, Section 2.2. For two subjects (4 and 5) we were able to visualize clinically optimized stimulation settings (determined by the clinician using a standard clinical approach) in relation to the Pareto set obtained from the subject-specific ground-truth monopolar (2D) models. A similar comparison could not be made for the remaining 3 subjects because one used a wide bipolar setting and one used double monopolar setting, configurations that were not tested during cEP/mEP recordings in these

subjects and so could not be accounted for in the models. For the third subject we did not have access to clinical programming data.

Both subjects had clinical stimulation settings with a C+/1- contact configuration and 60 µs pulse width, but at different amplitudes (3.6 mA and 2.25 mA, respectively). By estimating the Pareto Rank for the entire monopolar parameter space, we found that the monopolar stimulation setting selected by the clinician was contained within the first three Pareto ranks for subjects 4 and 5, representing 31% and 58% of the total parameter space, respectively (**Figure 6.5**). Across all subjects, the first three Pareto ranks represented a mean 39.2% of the parameter space with a range of [23.5, 58]%.

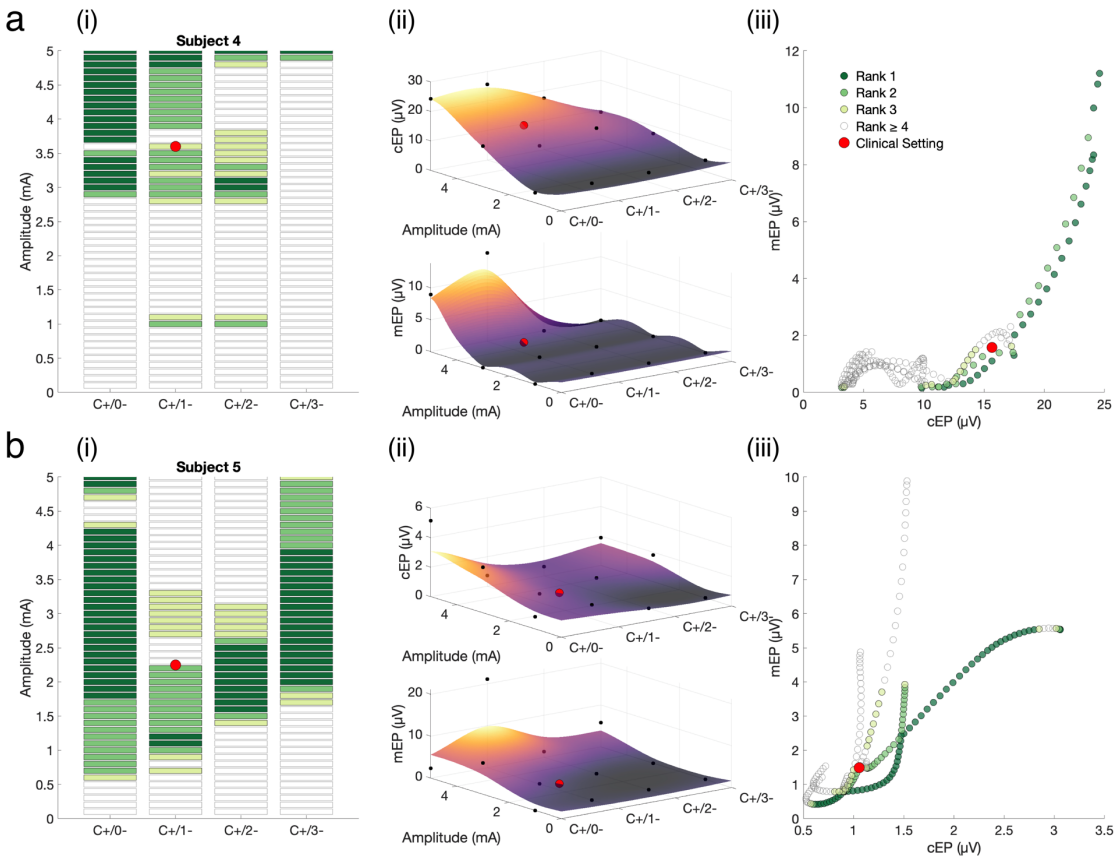

Figure 6.5: Clinical stimulation settings compared to the Pareto ranking from the ground-truth models.

(a) Subject 4, (b) Subject 5. Red circle represents the clinical stimulation setting. (i) the monopolar stimulation parameter space where the stimulation settings within the three lowest/best Pareto rankings are filled, and the stimulation settings with Pareto rankings higher/worse than three are left empty. (ii) The ground-truth models fit to the measured data (black dots). (iii) the objective space with the three highest/best Pareto rankings filled.

6.3.2 Multi-objective optimization for learning the pareto set

Next, we utilized the subject-specific ground truth models to assess performance of different optimization algorithms in their ability to accurately and efficiently estimate the Pareto ranking. To visualize the optimization process, we first present a representative example of the search trajectory at three different time points when optimizing over a monopolar (2D) parameter space (**Figure 6.6**). When only 5 samples (stimulation settings) are collected (**Figure 6.6**, *Left column*), there is a substantial mismatch between the two estimated models and the ground-truth, particularly the difference in the cEP and mEP magnitudes. This can be seen in the surrogate models fit to the simulated data (**Figure 6.6**, Top two rows) and in the estimated objective space (**Figure 6.6**, *Third row*). As a result, there is also a substantial mismatch between the Pareto ranking estimated using the surrogate models and the known ground-truth Pareto set (**Figure 6.6**, *Bottom row*).

As additional samples are collected (**Figure 6.6**, *Middle and right columns*), the surrogate models begin to better approximate the ground-truth. As a result, the Pareto ranking estimated from the surrogate models better aligns with the ground-truth Pareto set.

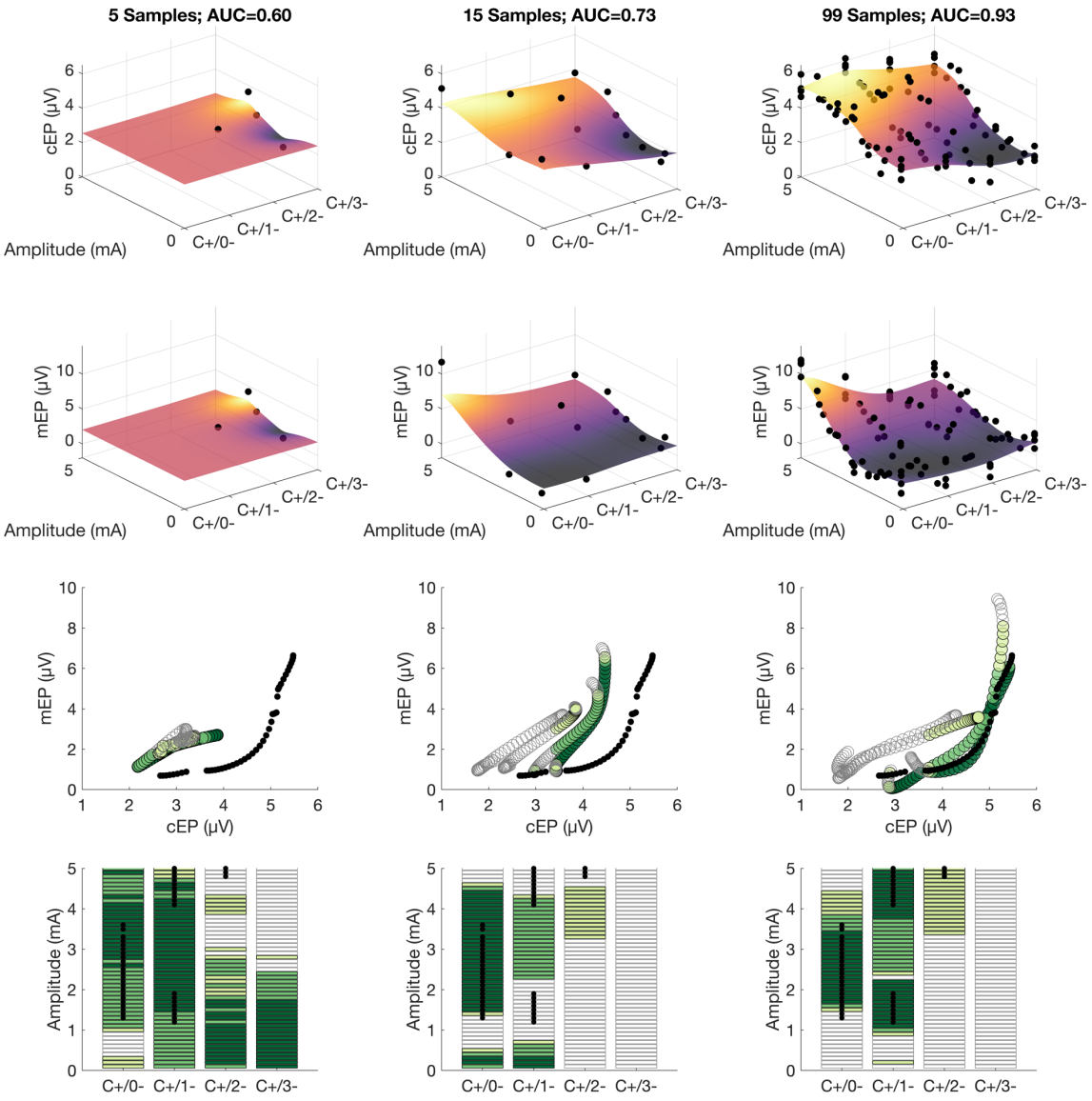

Figure 6.6: Search trajectory for EHVI+Model on a monopolar (2D) model.

Ground-truth model from Subject 1. Each column shows the state of the optimization algorithm after a certain number of samples have been collected. *Top two rows*, The surrogate cEP and mEP models constructed from the simulated data (black dots). *Third row*, The objective space estimated from the cEP and mEP models. Each circle corresponds to a stimulation setting from the full parameter space estimated with the cEP and mEP surrogate models. The filled green circles are those stimulation settings within the first three Pareto rankings estimated from the surrogate models. The black dots indicate the Pareto set calculated from the ground-truth models. *Bottom row*, The parameter space labeled with the estimated Pareto rankings.

6.3.3 Algorithm behaviour and performance

6.3.3.1 Algorithm component test

While the 10 combinations of search strategy and modeling approaches were evaluated for all three model parameter spaces – monopolar (2D), bipolar (3D), bipolar/pulse width (4D) – we first compare their performance on the bipolar (3D) models as they represent a common clinical scenario. Overall, the selection of both the search strategy and modeling approaches were significant predictors of performance in terms of final accuracy (search strategy: p-value = 7.9e-230; modeling approach: p-value = 0) and convergence rate (p-value = 2.4e-40 and p-value = 4.6e-66, respectively; **Figure 6.7a**). Between algorithms, those using the EHVI search strategy performed best in terms of final AUC and convergence rate. Next was the random search, which outperformed the evolutionary search strategy despite it using the surrogate model to guide sampling. The grid search strategy performed the worst. Between the two algorithms that could be used with or without a model, Grid and Random, the use of the surrogate model was critical to performing above chance (AUC > 0.50). The inclusion of a hyperprior did not improve performance for these search strategies, despite providing additional *a priori* information. On a per-sample basis (**Figure 6.7b**), the performance trajectory for EHVI+Hyperprior was better as early as 25 samples into the optimization and had a smaller variance across the 30 independent trials, when compared to the other search strategies combined with model hyperpriors. In the 5-way ANOVA model, trial was not a significant predictor of final AUC or convergence (p-value = 0.31 and p-value = 0.75, respectively).

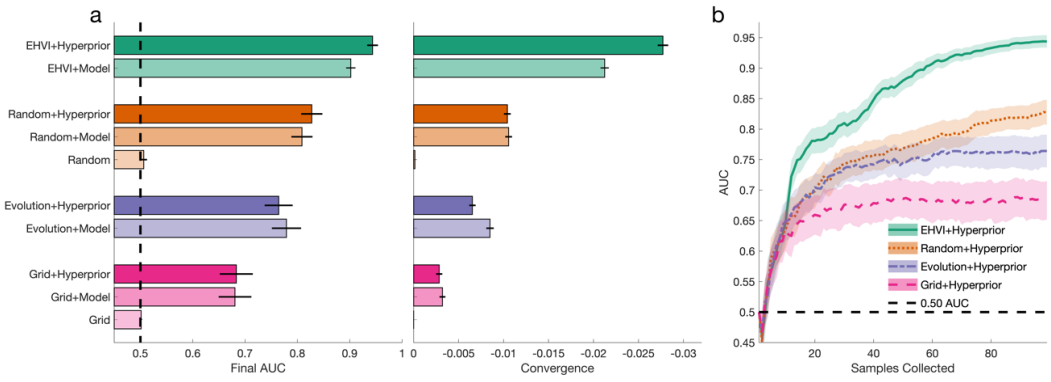

Figure 6.7: Comparison of algorithms on the bipolar (3D) parameter space.

(a) Final AUC and convergence for the 10 algorithm combinations. X-axis is reversed for the convergence rate as more negative values are preferable. Error bars indicate uncorrected 95% CI. (b) AUC trajectory measured at each sample. Patches indicate the standard error. Dashed line shows the chance threshold for AUC = 0.5. Performance for EHVI and EHVI+Hyperprior is shown when using the optimal configuration (Chapter 6, Section 3.3.3).

Table 6.2: Significance of predictors in 5-way ANOVA model.

Predictor/Interaction	Final AUC (p-value)	Convergence (p-value)
dimension[a]	**2.33E-306**	2.08E-20
subject[b]	**5.34E-72**	**7.47E-26**
model[c]	**0***	**4.59E-66**
search[d]	**7.88E-230**	**2.39E-40**
trial[e]	3.06E-01	7.48E-01
dimension x *subject*	**6.83E-43**	**7.71E-25**
dimension x *model*	**8.25E-147**	**2.44E-10**
dimension x *search*	**7.19E-28**	**3.98E-11**
dimension x *trial*	5.10E-02	4.33E-01
subject x *model*	**5.23E-37**	**1.18E-12**
subject x *search*	**3.05E-06**	**1.38E-06**
subject x *trial*	9.28E-01	1.86E-01
model x *search*	**3.95E-113**	**4.90E-22**
model x *trial*	5.74E-01	2.80E-01
search x *trial*	**4.24E-02**	9.47E-01

Values below 0.05 are bolded.

[a] dimensionality of the parameter space coded [0-2]

[b] subject data used to construct the model coded [0-3]

[c] modeling approach coded [0-2]

[d] search strategy coded [0-3]

[e] Trial number coded [0-29]

*p-value below MATLAB machine precision.

6.3.3.2 Parameter space dimension

After evaluating all algorithms on the bipolar (3D) ground-truth models, we compared the Hyperprior versions of each search strategy and characterized their performance across the monopolar (2D), bipolar (3D), and bipolar/pulse width (4D)

ground-truth models (**Figure 6.8**). We found that, in addition to search strategy, the parameter space dimensionality was a significant predictor of both final AUC (p-value = 2.33e-306) and convergence (p-value = 2.08e-20). As the dimensionality of the parameter space increased there was a general decrease in the final AUC achieved by each algorithm. However, this decrease was not uniform. The EHVI+Hyperprior algorithm showed no decrease in performance from 2 to 3 dimensions, but there was a significant (p-value = 1.5e-40) decrease in the final AUC for the Grid algorithm for each higher dimension. This ultimately resulted in the Grid+Hyperprior algorithm performing at chance on the bipolar/pulse-width (4D) ground-truth models. In between these two extremes were the Random+Hyperprior and Evolution+Hyperprior algorithms that performed equivalently to EHVI+Hyperprior on the monopolar (2D) models, but performed increasingly worse on the bipolar (3D) and then bipolar/pulse width (4D) models, with the performance of Evolution+Hyperprior decreasing more as the dimension increased. Finally, while EHVI+Hyperprior had similar final AUCs for the monopolar (2D) and bipolar (3D) models, the error trajectory showed that it converged much more rapidly on the monopolar (2D) models, and that only after 60 samples did it obtain the same accuracy on the bipolar (3D) models.

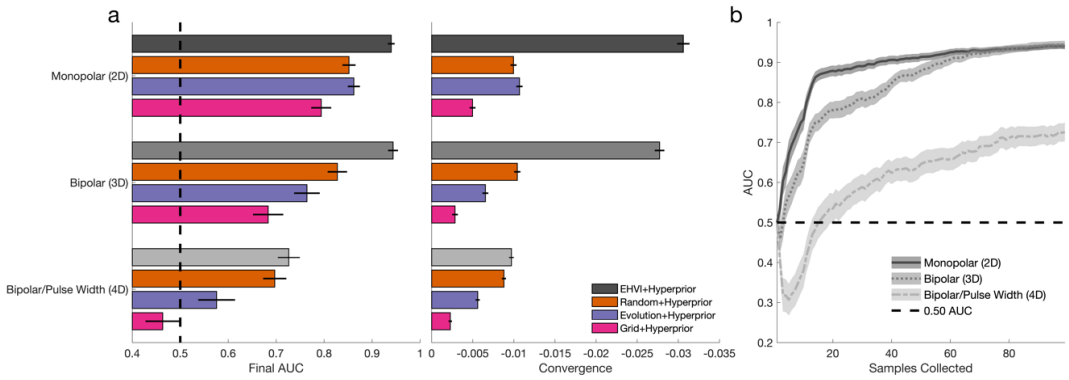

Figure 6.8: Search strategy performance across different dimension parameter spaces.

(a) Same as in **Figure 6.7**. (b) Sample-wise AUC across all trials and subjects for models of a given dimension. Performance for EHVI+Hyperprior is shown when using the optimal configuration (Chapter 6, Section 3.3.3).

6.3.3.3 EHVI and Hyperprior configuration parameters

The overall behavior of the EHVI+Model and EHVI+Hyperprior algorithms could be further modified by adjusting two internal configuration parameters: κ, which controlled the exploratory behavior of the algorithm, and λ, which determined to what extent the hyperprior information was taken into account when fitting the surrogate model, and therefore was not relevant to EHVI+Model. For each of the subjects, the performance of the EHVI+Model algorithm was evaluated for different values of κ, and the EHVI+Hyperprior evaluated for different values of κ and λ. With EHVI+Model there was a clear improvement in both the final error and convergence rate for $\kappa = 10$, while performance was slightly worse for higher values, and significantly so for lower values (p-value = 1.74e-46). (**Figure 6.9a**). There was also a larger variation in the AUC trajectory when κ is low (**Figure 6.9b**) as the lack of exploration led to premature convergence. In

the EHVI+Hyperprior algorithm, varying κ had a similar effect as in EHVI+Model, (p-value = 1.00e-23), while varying the hyperprior configuration parameter, λ, did not significantly affect the overall results (p-value = 0.853, linear regression) (**Figure 6.9c-d**).

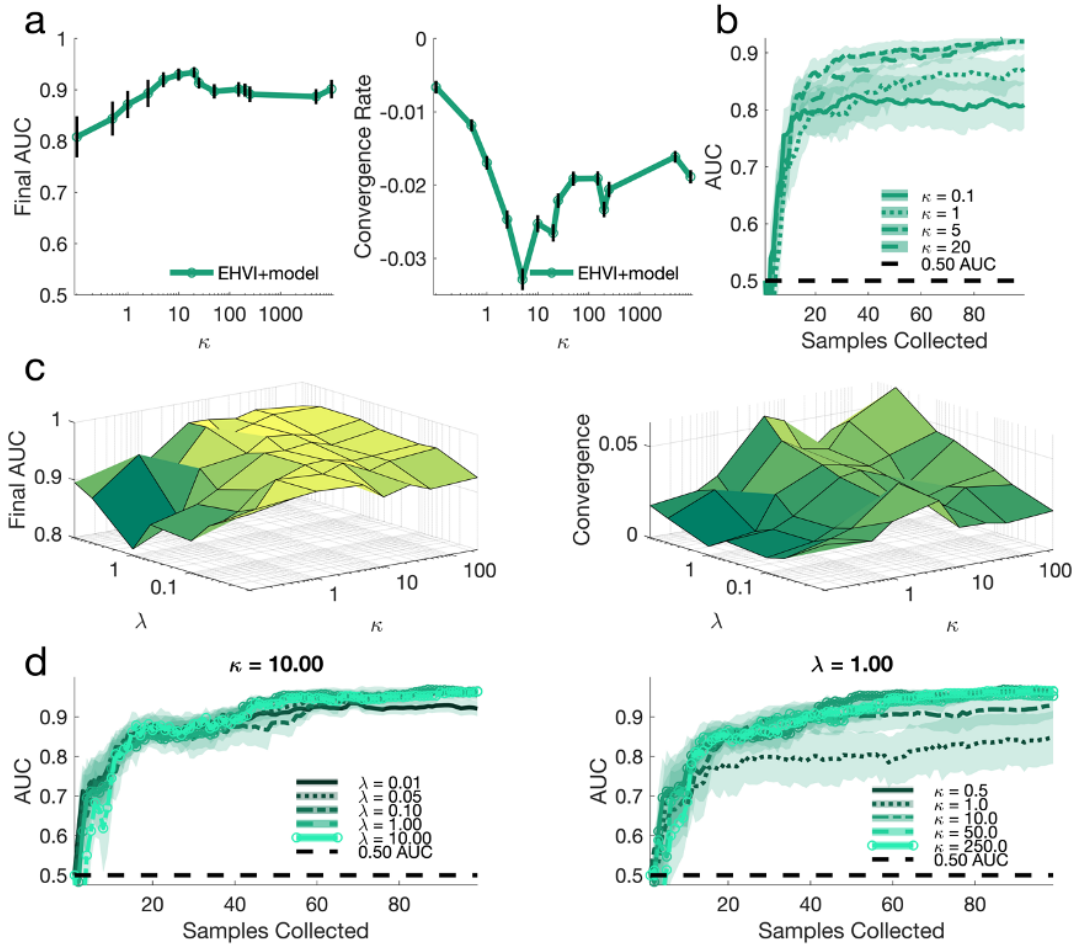

Figure 6.9: Effect of EHVI configuration parameters on performance.

(a) Final AUC and convergence for the EHVI+Model algorithm as the configuration parameter kappa is varied. (b) AUC trajectory for select values of κ. (c) Surfaces shows the final AUC and convergence as both kappa and lambda is varied. (d) AUC trajectories for (*Left*) select values for λ with $\kappa = 10$, and (*Right*) select values of κ with $\lambda = 1$.

6.3.3.4 Computation time

Across the different models, the three algorithms that performed the best were EHVI+Hyperprior, EHVI+Model, and Random+Hyperprior. However, the first two had non-negligible computation times for each sample. With a limited time-window to perform the experiments during surgery, spending excess time to determine the next sample would reduce the total number of samples that could be collected. Therefore, the increased performance of a more sophisticated algorithm must offset any additional computational burden. By assuming a 25 minute (100 samples at 15 seconds per samples) window to perform the experiments and accounting for the three-second washout period between samples already built into the experimental protocol, we can recalibrate the algorithm performance within that limited time-frame (**Figure 6.10**). This recalibration had no notable effect on the algorithms when applied to the monopolar (2D) models. However, there was a significant decrease in the performance of the EHVI+Hyperprior algorithm for both the 3D bipolar (3D) (p-value = 0.0073) and bipolar/pulse width (4D) (p-value = 5.13e-05), compared to EHVI (p-value = 1.000 and p-value = 0.384, respectively).

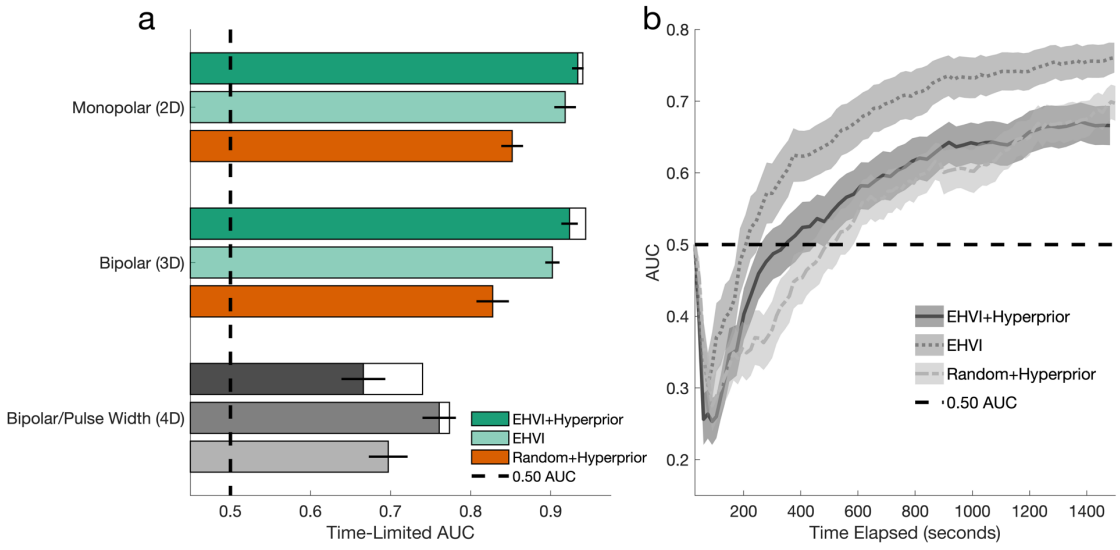

Figure 6.10: Wall-time comparison between EHVI+Hyperprior and Random+Model.

(a) Final AUC for the top three algorithms when corrected to a maximum of 25 minutes wall-time. Error bars indicate 95% CI. Open boxes show the AU-ROC when all 100 samples are included. (b) Sample wise AUC trajectory for all subjects and trials using the bipolar/pulse width (4D) models. X-axis has been scaled to reflect the computation time for the algorithms, rather than the number of samples. The gray bars for Time-Limited AUC in (a) correspond to the AUC trajectories over Time-Elapsed in (b).

6.3.4 Effect of objective function characteristics on performance and behaviour

Just as changes to the components and configuration parameters can affect the performance and behavior of an algorithm, so can the properties of the optimization problem it is trying to solve. Here we quantify how different properties of the optimization problem – subject-to-subject heterogeneity, and noise of the objective function – influence a representative set of optimization algorithms.

6.3.4.1 Variability between subject-models

Comparing the final AUC and convergence between individual subjects, the general trends described in Chapter 6, Section 3.3.1. are preserved (**Figure 6.11a**). However, there is significant variability in final AUC and convergence between the subjects (p-value = 5.34e-72 and p-value = 7.47e-26, respectively; 5-way ANOVA) For the more sophisticated EHVI+Model and EHVI+Hyperprior algorithms the final AUCs are comparable, with EP010 trending lower than the other three (p-value = 9.09e-29). But, for the less complex Random, Evolution and Grid search strategies, the final AUC for Subjects 3 and 4 are significantly lower than for Subjects 1 and 2, creating two groups of subjects with apparently 'easy' and 'difficult' objective functions. While the difference between the two groups is not clear for the EHVI+Hyperprior algorithm after 100 samples, the overall trajectory shows how the AUC increases much faster for the 'easy' subjects than for the 'difficult' subjects (**Figure 6.11b**).

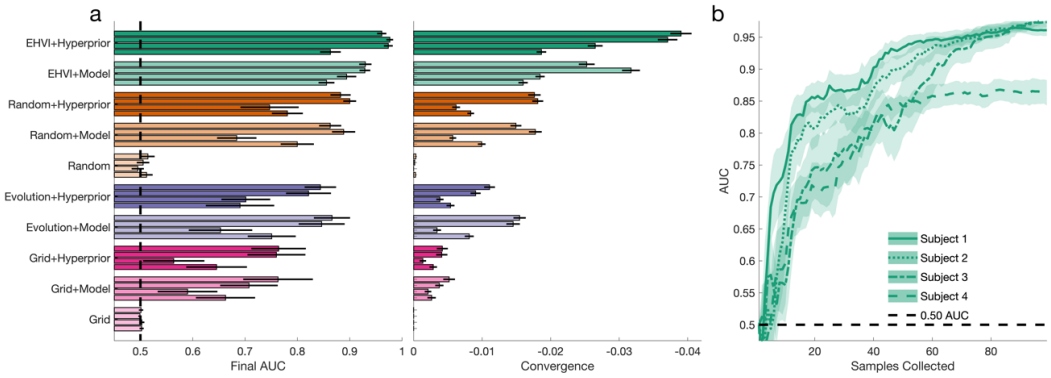

Figure 6.11: Algorithm performance on individual bipolar (3D) subject models.
(a) Final AUC and convergence after 100 samples for the four subjects with bipolar (3D) stimulation data. (b) Sample wise AUC trajectory for the EHVI+Hyperprior algorithm on individual subjects. Error bars (a) and patches (b) indicate 95% confidence interval.

6.3.4.2 Length scale

To identify the differences between the 'easy' and 'difficult' objective functions we consider the length scale hyperparameters of the GP model fit to the subject data. As described in Chapter 6, Section 2.2.3 (Eq. 5), the length scale hyperparameters determine how much the output of the objective function can vary between adjacent inputs [26]. At lower length scales, the output can vary more, so we would expect a 'wigglier' objective function that may be more difficult to optimize. By comparing the length scales for the 4 bipolar (3D) models to the final AUC across the different optimization algorithms, we found that final AUC was significantly correlated (Spearman, p-value < 0.05) with the length scales corresponding to the cathode and anode parameters of the cEP models, and the cathode parameter of the mEP models (**Figure 6.12**). This indicates that more complex objectives, as measured using length scale, are more difficult to optimize, as measured by final AUC.

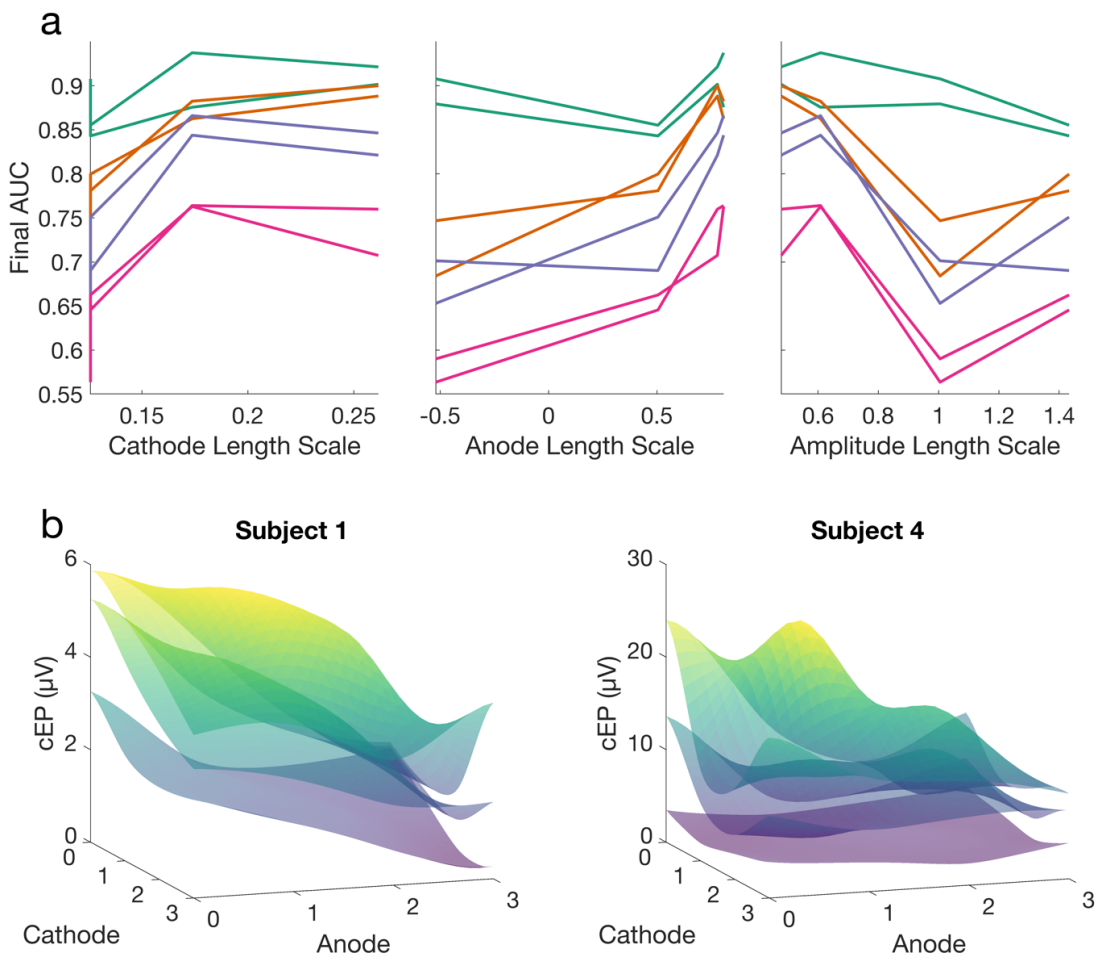

Figure 6.12: Effect of length scale hyperparameters on performance.

Each line shows the final AUC for each algorithm across each subject as a function of the length scale hyperparameters. (a) Length scales of GP model fit to cEP data. (b) Bipolar surfaces for the subjects with the lowest and highest length scales for the cathode and anode parameters. The surfaces within the plot represent the cross-sections at 1, 3, and 5 mA stimulation amplitude.

6.3.4.3 Measurement noise

Lastly, measurement noise can vary between different subjects, recording equipment, and changes to the biomarker, so it is valuable to know the tolerance of an algorithm to per-sample variability. To test this, the contrast-to-noise ratio (CNR) was

artificially scaled to produce objectives with more and less measurement noise, and the performance of the four search strategies was evaluated (**Figure 6.13a**). As the CNR decreases, the performance of the optimization algorithm monotonically decreases and the variability of the AUC increases. These curves show that most algorithms can tolerate an 2x decrease in CNR without much degradation in performance, and can perform better than chance (AUC = 0.50) at up to a 10x decrease in CNR.

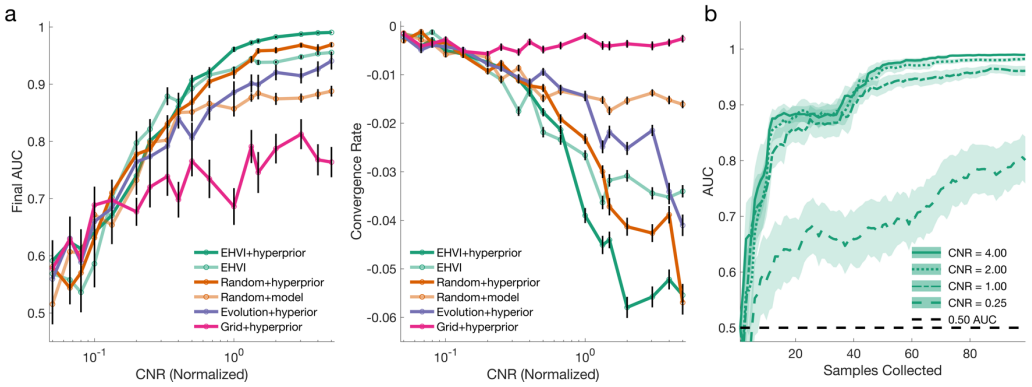

Figure 6.13: Effect of noise on performance.

(a) Each line represents the final AUC for a given algorithm as the measurement noise of the objective function is altered. (b) Per-sample trajectory of the AUC for the EHVI+Hyperprior algorithm on a subset of objectives with different measurement noise. Error bars and patches indicate the standard error.

6.3.4.4 Optimizing bipolar/pulse width (4D) parameter space for 500 samples

Given the limited time available to perform these experiments in the OR, most of our analysis was limited to 100 samples or fewer. In the case of the bipolar/pulse width (4D) model, this resulted in no algorithm achieving a final AUC greater than 0.7. To better characterize the performance and behavior of these algorithms, we performed an additional simulation experiment that was allowed to run for 500 samples (**Figure 6.14**). After 500

samples, the model-based Random and EHVI search strategies achieved a final AUC greater than 0.8. This shows that these algorithms can scale to higher dimensions, but will need additional samples to achieve performance comparable to the bipolar (3D) models.

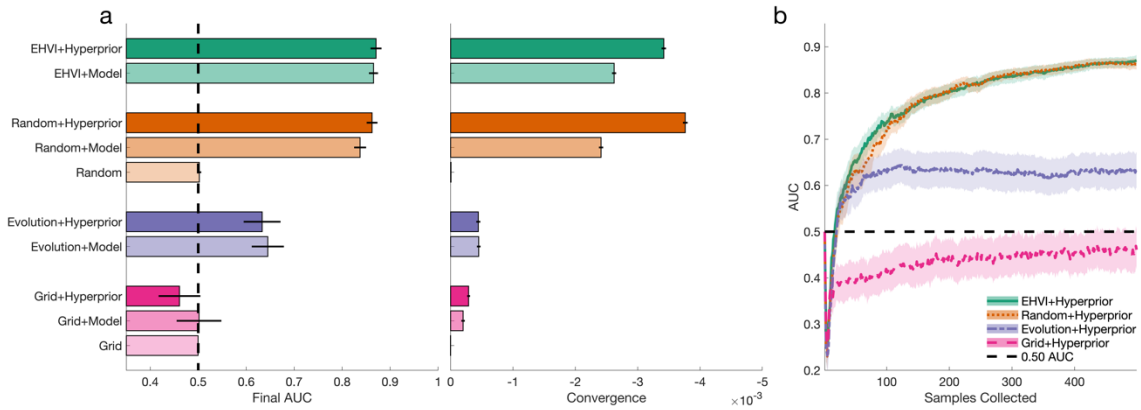

Figure 6.14: Performance for optimization trials with 500 samples.

(a) Final AUC and convergence after 500 samples for the four subjects with bipolar/pulse width (4D) stimulation data. (b) Sample wise AUC trajectory for each search strategy with Hyperprior algorithm on individual subjects. Error bars (a) and patches (b) indicate 95% confidence interval.

6.4 Discussion

We used electrophysiology data (cEP and mEP) recorded from patients with PD undergoing STN DBS surgery to create a platform for evaluating performance of multi-objective optimization algorithms designed to reduce the size of the stimulation parameter space by estimating the Pareto ranking. While data-driven optimization approaches have been proposed for designing or adjusting stimulation settings for neural modulation, this is the first time that multi-objective data-driven optimization (maximizing cEP and minimizing mEP) has been applied to direct electrophysiological biomarkers in a clinical neural modulation application. Among those we evaluated, the best algorithm for this

purpose was a sample-efficient Bayesian optimization approach with the EHVI acquisition function. Critical algorithm components included the use of a surrogate model, and a well-configured acquisition function. We demonstrated that application of our optimized algorithm reduced the parameter space available to the clinician by at least a third, although this study was not designed to validate the use of biomarkers for clinical programming. This study followed a framework for designing multi-objective data-driven optimization solutions that can be applied across DBS applications and electrode designs [37].

6.4.1 Biomarkers for optimizing DBS

The search for DBS biomarkers that can serve as a surrogate to clinical observation is an active area of research. In this study we used cEP and mEP to highlight the importance of multi-objective optimization (maximizing benefit, minimizing side effects). The most well-studied biomarker in PD is subthalamic beta power, which has been investigated as a signal for responsive control [72], [74] of stimulation parameters in NHP models of PD, and in small studies of responsive stimulation in humans [71], [75]. Cortical gamma power has been proposed as a biomarker of dyskinesias (a side effect) [17]. Non-electrophysiologic biomarkers of DBS efficacy include kinematic measurements [27], which can serve as a surrogate for tremor severity, and thereby used as an objective function for closed-loop optimization techniques in patients with PD or essential tremor [88], [89].

Beyond PD and movement disorders, a recent study in DBS for epilepsy found that in patients with bilateral electrodes in both the anterior nucleus of the thalamus (ANT) and

hippocampus, that the ability to elicit hippocampal EPs through ANT stimulation was correlated with a decrease in seizure frequency [90]. As they become better understood, each of these biomarkers can be incorporated as an objective for a single- or multi-objective optimization system.

6.4.2 Prototyping data-driven multi-objective optimization

By using the cEP and mEP data in combination with the previously reported framework [37] for designing data-driven optimization systems, newly extended for multi-objective optimization problems, we were able to evaluate how 10 different algorithms performed on three different parameter spaces across five subjects (one subject had only monopolar (2D) parameter space). From the component test, it was clear that the most important feature for this optimization problem was the use of a surrogate model. Without using a surrogate model to interpolate across the entire parameter space, the Pareto ranking could only be calculated for the stimulation settings that were actually tested by the algorithm. While this approach may work when many samples can be collected relatively quickly it does not work well with the limited data that can be collected during this particular experiment. In contrast, when using a surrogate model to interpolate the effect of untested stimulation settings on cEP and mEP, the Pareto ranking could be estimated across the entire parameter space. Moreover, the surrogate GP models allowed the noiseless measurements of cEP and mEP to be estimated.

Second in importance was the use of the EHVI acquisition function, especially on the higher dimensional parameter spaces. While we only considered one acquisition

function, this is an active field of research and it is certain that even more effective acquisition functions can be integrated with this system [91]–[93]. Hyperpriors also did offer some benefit, but this was offset by the increased time necessary to calculate the next sample. However, this can be addressed with high-performance computing hardware and software-level changes. Moreover, this does not diminish the potential utility of alternative optimization approaches where data from previous subjects are used to inform the surrogate model fitting or search algorithm.

The behavior of the optimization algorithm is inexorably connected to the characteristics of the optimization problem. One of the key characteristics of the objective function is the dimensionality of the parameter space. As we compared the same algorithms across parameter spaces of increasing size/dimension, we found that they all suffered a decrease in performance. However, the performance of the algorithms deteriorated at different rates. In the case of the Grid algorithm, even when using a surrogate model, it was unable to perform better than chance on the bipolar/pulse width (4D) parameter space. When the simulations were allowed to run for 500 samples on the bipolar/pulse width (4D) models (6.3.4.4), the Grid algorithm still performed below chance. However, after 50 samples the EHVI+Hyperprior and Random+Hyperprior algorithms achieved AUCs on the bipolar/pulse width 4D parameter space comparable to the bipolar (3D) parameter space. While these methods can be directly extended to more complex electrodes with higher-dimensional parameter spaces, these findings show that it will take a combination of the right algorithm, and more efficient data collection to solve these more difficult optimization problems.

6.4.3 Related and future work

Data-driven optimization approaches have been gaining traction as an effective tool for designing and optimizing stimulation settings for neural modulation. Recent approaches have used multi-objective particle swarm optimization to identify stimulation settings that selectively activated fiber tracts in subject-specific biophysical/finite-element models [94], and a genetic algorithm to optimize a thalamocortical basal ganglia model-based proxy of stimulation effectiveness [29]. While powerful, these approaches rely on complex models that are time-consuming to construct, and evolutionary optimization algorithms that can often require >10,000 samples to converge, limiting them to offline use. An alternative approach that could be applied online used a single-objective Bayesian optimization algorithm to tune a controller for regulating STN beta power in a mean-field model of PD [65]. Beyond DBS, data-driven optimization has also been applied for spinal cord stimulation [95]. In the approach presented here, we leveraged a sample-efficient Bayesian optimization approach to solve a difficult multi-objective optimization problem based on electrophysiological biomarkers that can measured in real-time. Taken together, this will pave the way for subject-specific online optimization of multiple objectives *in vivo*.

While the EHVI+Hyperprior Bayesian optimization algorithm outperformed all others in this study, it will still be necessary to validate *in vivo*. However, there are several additional engineering hurdles that need to be overcome, including automated real-time calculation of the cEP and mEP features. If a Random or Grid search strategy had performed best, the cEP and mEP measurements could be done offline. However, since our

analysis showed improved performance with a Bayesian optimization approach that relies on real-time measurements, these efforts are justified. Additionally, as 8-contact segmented leads are becoming standard, the surrogate models will need to be adapted to this more complex parameter space. The parameter space of bipolar contact configurations on a segmented lead can reasonably be represented in four dimensions by using x-y coordinates as parameters for the cathode and anode, plus parameters for amplitude and pulse width. The GP models used by the Bayesian optimization algorithm here could be readily applied to such a 6D parameter space. However, if stimulation settings were to become complex enough that ten or more dimensions were needed to define the parameter space, other models specifically designed for high-dimensional Bayesian optimization should be considered [67].

In parallel to the implementation of this data-driven optimization approach *in vivo*, there are several natural extensions to the algorithm itself. One such extension is the inclusion of energy consumption as an objective to minimize. This can be valuable, as many DBS devices require an outpatient surgery to replace the battery and patients with rechargeable devices would still benefit from an increased battery life [96]. Another option is to incorporate optimization with learnable safety constraints [58]. These algorithms are designed to avoid regions of the parameter space that could be dangerous. While stimulations that create unwanted mEP responses are not dangerous *per se*, avoiding these stimulation settings could further refine the parameter space to improve efficiency and provide a more comfortable experience for patients. Finally, state-dependent optimization [48] can be used to learn a policy for adjusting stimulation based on the underlying neural

state – for example, as medication levels change. As each of these approaches introduce additional complexity to the optimization problem, it will still be necessary to properly design the algorithms by following a framework such as described here.

6.4.4 Limitations

The conclusions drawn from this study should be limited to the performance and behavior of the data-driven optimization algorithms, and not the clinical implications of the stimulation settings or the biomarkers, as they have not been validated. Additionally, the datasets for some subjects were incomplete, which required us to augment the dataset with additional points to provide robust ground-truth models for simulation experiments. While the augmented data were based on experimental findings that were consistent across patients (e.g. EP amplitude sigmoidally increasing with increasing pulse width), it still likely resulted in ground-truth models less consistent with the original subjects. However, the overall characteristics of the models still allowed for evaluation of the algorithms. Additionally, with the data available, we could not compare clinical settings with the calculated Pareto ranking for all patients. But this was not the focus of the current study and only served as a demonstration of the potential utility of a multi-objective approach. A prospective study would be necessary to evaluate performance of optimization algorithm and putative biomarkers for clinical programming. Finally, our ground-truth models assumed that the biomarker response to stimulation was stationary – i.e. the effect of brief stimulation does not change over time. Given that cEP and mEP quantify direct activation of relevant neural pathways, and the consistent repeated stimulation results in Chapter 6, Section 5.1, this is reasonable but warrants further investigation.

While the ultimate goal in DBS programming is to find the single stimulation setting that best treats the patient, our multi-objective approach produces a reduced set of potential settings. Even though electrophysiological biomarkers, such as cEP and mEP, maybe be quantitative and more objective, they cannot account for all aspects of the therapy and some degree of clinical testing will be necessary. This approach would allow that clinical testing to focus on stimulation settings with a higher likelihood of being effective.

6.4.5 Conclusion

Data-driven multi-objective optimization approaches have the potential to improve DBS programming and provide patients with more precise and effective therapies. However, these optimization algorithms need to be carefully designed, as the wrong choices can lead to chance-level performance, or take too long to compute, therefore not saving time in the long run. We describe a framework based on Bayesian optimization that is well suited to this problem. Future studies are needed to test the utility of optimization algorithms in real-time and assess their performance with alternative biomarkers or objective functions.

6.5 Supplement

6.5.1 Estimating measurement noise

To provide a better estimate of the sample-to-sample measurement noise, data from an additional Subject 6 was collected where some stimulation settings were applied twice. Monopolar stimulation at 3 mA was applied at each of the 8 segmented contacts on an

Abbott 6172 clinical electrode twice during the data collection experiment. These repeated stimulation settings and cEP were fit to a model as described in Chapter 6, Section 2.2 (**Figure 6.15a**), and the contrast to noise ratio (CNR) was calculated. The noise in the simulation models for subjects 1-4 was then calibrated to that CNR.

Repeated stimulation using identical settings produced consistent cEP responses, with a CNR of 8.2. Additionally, no trend was observed when comparing the first stimulation to the second stimulation using the same setting (**Figure 6.15d**). This indicates that the order and timing of stimulation does not influence the effect of the stimulation on cEP.

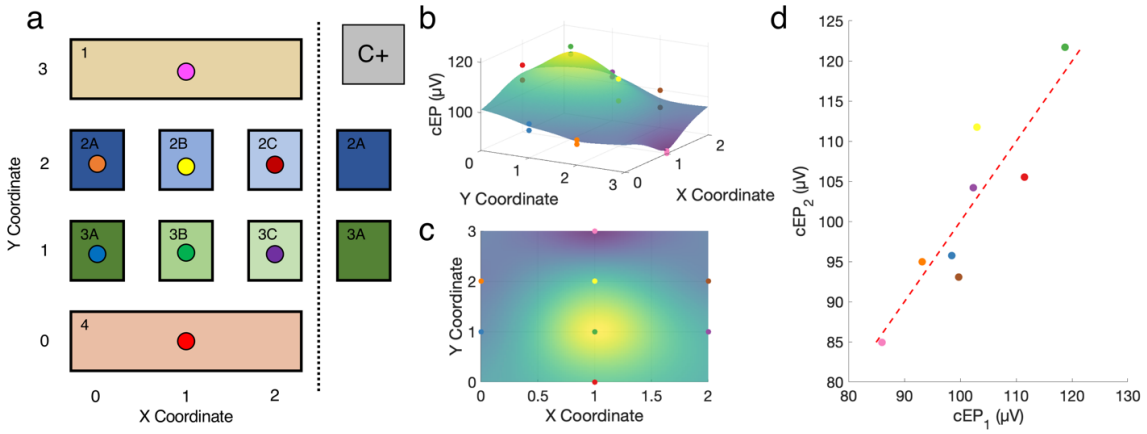

Figure 6.15: Repeated stimulation in a segmented electrode to estimate measurement noise.

(a) Spatial layout of the monopolar parameter space for a segmented electrode. Each colored circle corresponds to a stimulation setting (b) GP model fit to the repeated stimulation settings. The surface is the expectation of the GP model. (c) Top down view of the GP model. (d) Comparison between the first and second applications of the same stimulation setting. Dashed red line is the equality of the two measurements.

CHAPTER 7. CONCLUSION

This dissertation presented a framework for designing data-driven optimization systems for neural modulation. This framework was developed using a medial septum optogenetic stimulation model with the objective of maximizing hippocampal gamma, the power of the oscillations between 33 and 50 Hz measured from the hippocampus. The framework was then extended to a number of other neural modulation applications including, optimizing to a setpoint, state-dependent optimization, optimization with learnable safety constraints, and optimization with multiple objectives. This dissertation covers these extensions through different components of the framework, but efforts are ongoing to build upon these results (**Figure 7.1**).

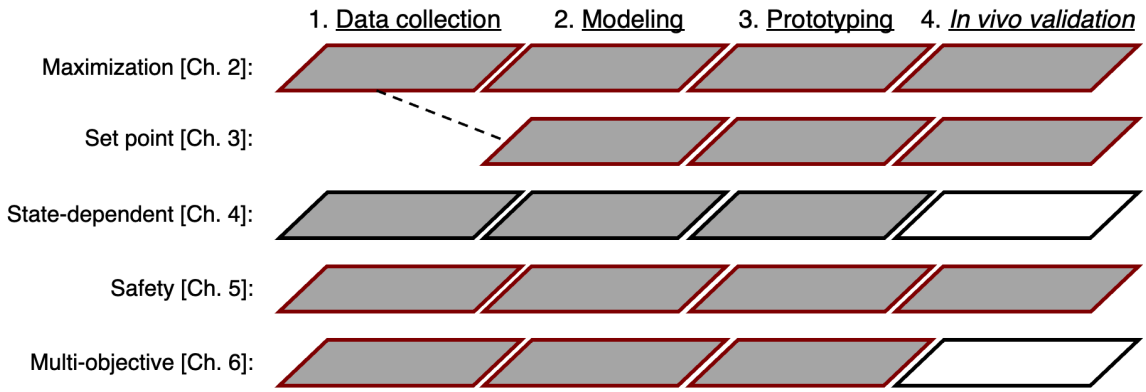

Figure 7.1 Status of each project based on the framework.
Filled bars indicate each step of the framework covered in this dissertation for each project. Bars with red borders indicate completed work, and black borders indicate a proof-of concept.

The following sections contextualizes the different projects in terms of the practical considerations when applying the framework, and role and utility of the framework for engineering, scientific, and clinical perspectives.

7.1 Implementing the framework in practice

7.1.1 Data collection

The first step to implement this framework is the collection of data from an initial cohort of subjects. Since the goal is to use the data from subjects 1 through N to design an effective algorithm for some future, unknown subject N+1, it is important that samples collected are representative of the entire parameter space. The approaches for collecting the data can be loosely categorized as non-active and active.

7.1.1.1 Non-active data collection

Non-active data collection includes approaches where the set of stimulation settings to be evaluated can be defined in advance. The simplest approach is a grid search where a limited number of values are selected for each parameter and all combinations are evaluated. For example, in a 2-dimensional parameter space of amplitude and frequency bounded on 0-4 mA and 20-200 Hz, respectively, with 2 values for each, a possible set of stimulation settings could be [1 mA, 20 Hz], [3 mA, 20 Hz], [1 mA, 130 Hz], and [3 mA, 130 Hz]. This is the approach used for data collection in the rat optogenetic medial septum studies and the human STN DBS experiments. While adequate for parameter spaces with few dimensions, it does not scale well to higher dimensions.

An alternative to the grid search is a uniform random search [97]. In this approach, stimulation settings are selected by drawing a random value for each parameter from independent uniform distributions. Using the same parameter space defined in the grid search example, a possible set of stimulation settings could be [3.8 mA, 84 Hz], [1.9 mA, 183 Hz], [3.2 mA, 158 Hz], and [0.8 mA, 22 Hz]. The advantage of this approach is that it evaluates 4 different values for amplitude and 4 different values for frequency, compared to just 2 each for the grid search. While still useful for building regression models, one limitation of a uniform random search is that the data is not amenable to classical statistical tests like analysis of variance. Regardless, a uniform random search can be an effective approach for many of the parameter spaces possible with commonly used neural modulation technologies.

7.1.1.2 Active data collection

While effective at lower dimensions, the grid and random search strategies do not scale to the higher dimensional parameter spaces becoming possible with new developments in neural modulation technology with many additional parameters and settings to optimize. As such, more sophisticated approaches will be needed for collecting the data needed in this design framework. One natural approach is active learning. Whereas the goal of an optimization algorithm is to identify a global maximum of an objective function by avoiding suboptimal regions of the parameter space, the goal of an active learning algorithm is to learn the objective function across the entire parameter space. In addition to the many dedicated active learning algorithms, the Bayesian optimization

approach used in this framework can be framed as active learning by using an acquisition function to minimize uncertainty, but not to optimize [36].

Another situation that could potentially require active data collection would be in a clinical context, where it may not be ethical to intentionally submit subjects to suboptimal stimulation settings. In such a case, an optimization approach combining a high level of exploration along with learnable safety constraints could be used to carefully explore the parameter space within a therapeutically reasonable window. While not motivated by clinical constraints, much of the data used to design the safe optimization algorithm in Chapter 5, was collected during previous optimization experiments, but was still useful within the design framework.

7.1.2 Modelling

The modeling step is where the data collected from an initial cohort of subjects is used to construct simulation models that can act as proxies for the actual subjects. The key advantage of the data-driven optimization approach is the minimal number of assumptions needed for building the simulation models. Rather than modeling the underlying biophysical mechanisms of stimulation and the brain, statistical models can be used to directly model the data. Since a data-driven optimization algorithm is not concerned with the underlying mechanisms, they do not need to be accounted for by the simulation model.

7.1.2.1 Model structure

When selecting the model structure, it is important that it matches the characteristics of the response across the parameter space. Particular considerations include, the linearity of the response and the presence of discontinuities, along with the distribution and heteroscedasticity of the noise. In the simplest case, a polynomial with Gaussian noise may suffice. However, nonlinear models with more esoteric noise distributions may be required. In all of the examples shown here, the data was modeled using a Gaussian process regression model. This non-parametric approach was sufficient for capturing the relatively smooth changes in the response to stimulation across the parameter space and was able to naturally estimate the distribution of the noise. In practice, for neural modulation problems with fewer than 10 dimensions and no sharp discontinuities, Gaussian process regression models with squared exponential or Matérn covariance functions are a reasonable first approach.

However, GPs do not scale well with large amounts of data (>1000 observations) or to higher dimensions (>10), as inference will take an extremely long time, decreasing the throughput for testing different data-driven optimization algorithms and configurations. In these situations, model structures such as random forests or neural networks might be more appropriate.

7.1.2.2 Data augmentation

Another aspect of constructing the models is data augmentation. While the ideal case would be to prospectively collect data from the initial cohort such that it covers the

entire relevant region of the parameter space, this may not always be possible. For example, if the initial data was collected as part of a different study, regions of the parameter space may not have been extensively sampled, but may be relevant for the purpose of the optimization problem. In this case, the data set can be augmented with additional data points based on prior knowledge or assumptions to construct a more physiologically realistic model. In Chapter 5, when constructing models mapping stimulation amplitude to performance on a memory task, additional data points with poor performance were added at higher voltages to create an unsafe region of the parameter space without the need to expose the subjects to unsafe stimulation settings. In Chapter 6, not all combinations of the 4 stimulation parameters were tested in all subjects due to limited time available to collect data during the surgery to implant the DBS device. In this case, the augmenting data was estimated based on the other subjects and other experiments. While the resultant simulation models were likely less accurate proxies of the original subjects, the overall characteristics, such as noise and complexity, made them sufficient for prototyping and benchmarking different data-driven optimization approaches.

7.1.2.3 Subject characteristics

A final consideration when deciding the model structure is what other information can be interpreted from the relationship between the characteristics of a given model and the performance of different algorithms. A metric that can applied to essentially any model is the CNR. Borrowed from image processing, this metric was significantly correlated with the performance of the optimization algorithms in Chapter 2, Section 5.5 and Chapter 6, Section 3.4.3. In other words, the lower the CNR, the more challenging the optimization

problem. In Chapter 5, the behavioral objective in the memory optimization experiments had a CNR below 1.0 for some subjects. This explains some of the challenge of that optimization problem, despite only optimizing a single parameter. In contrast, the evoked potential objectives in Chapter 6 had a CNR above 8.0, which allowed for optimization of a 4 dimensional parameter space using a more complex multi-objective optimization approach.

Another characteristic of the objective function that can be interpreted from the Gaussian process models used in this dissertation are the length scales. The Gaussian process covariance functions with automatic relevance determination fit a length scale hyperparameter to each dimension/parameter of the parameter space that controls how much the output can change between two adjacent stimulation settings. In Chapter 6, optimization algorithms performed worse in the subjects with the smaller length scales. In this case the relative complexity of the different subjects could be readily observed. However, with additional subjects or higher dimensional parameter spaces, a quantitative measure will be useful for assessing the difficulty of the optimization problem, measuring the relevance of each dimension/parameter, and for identifying clusters of similar patients.

7.1.3 Prototyping

After the construction of the simulation models that can serve as high-throughput proxies for the original subjects, the next step of the framework is to use these simulation models to benchmark different data-driven optimization algorithms.

7.1.3.1 Selection of algorithms and configurations

In Chapter 2, Bayesian optimization, a surrogate model algorithm, outperformed the gradient approximation and evolutionary algorithms. This was generally expected based on the characteristics of the optimization problems – smooth, noisy, expensive – but it was important to verify this empirically. However, Bayesian optimization is not suitable for every problem. In Chapter 2, Section 5.5, the UCB(0.4) algorithm that performed well on the optogenetic optimization problem regularly failed to find the optimum on the rare-event Easom benchmark objective function. In objectives with rare or many local extrema, discontinuities, or high-dimensional parameter spaces, an evolutionary algorithm may have better performance. Ultimately, these characteristics are determined through the data collection and modeling steps of the framework, and the appropriate algorithm is determined empirically through prototyping.

Another consideration in the selection of algorithms is the execution or update time. This is the amount of time the algorithm needs to propose a new set of stimulation settings after receiving the output from the previous iteration. This was not an issue in the single objective optimization problems as the update time was generally less than a second, and iterations of the algorithm was seconds to days apart. However, in Chapter 6, some of the multi-objective optimization approaches took a considerable amount of time to update, such that it impacted the overall performance of the algorithm given the limited time with which data can be collected. This issue may be addressed through the use of additional computing resources or modifications to improve the efficiency of the algorithm.

7.1.3.2 Design criteria

In conjunction with selecting the algorithms to prototype, the performance metrics to evaluate them must be specified. The primary design criterium for almost any single objective optimization problem is the error between the output of the underlying objective function at the estimated optimal and the ground-truth optimum. However, in the case of multiple objectives, the output of the optimization algorithm is a Pareto set of stimulation settings. As a result, measuring the error between the estimated optimum and the ground-truth solution is less straightforward. In Chapter 6, the output of the multi-objective algorithm was framed as a classification of whether a stimulation setting was within the Pareto set or not. While this worked for the goal of reducing the overall size of the parameter space, other metrics may be more appropriate depending on the specific goals of the optimization problem.

In addition to the overall accuracy of the estimated optimal, there are other secondary criteria that can be used to assess the performance of the algorithm. For any measure of error, the convergence rate can be calculated. However, it should be noted that convergence rate describes how quickly the algorithm arrived at *a* solution, not necessarily the accuracy of the solution. An alternative approach to combine error and convergence rate is the sum of the error over the entire optimization trajectory [98]. While error and convergence rate can serve as general purpose performance criteria, other optimization problems may require specific solutions. For example, when designing the optimization algorithm with learnable safety constraints in Chapter 5, the maximum overshoot of the amplitude into the unsafe region of the parameter space was used as a problem-specific

performance metric to evaluate safety of the different algorithm configurations. Similarly, in Chapter 6, the computation time to select the next sample to evaluate was also considered as part of the design criteria.

7.1.4 In vivo validation

7.1.4.1 Verifying assumptions

The key purpose of *in vivo* validation is verifying the assumptions made during modeling. For example, in all of the examples shown in this dissertation, it was assumed there was no interaction between the effects of stimulation. Consequently, in most of the examples, the data collection procedure randomized between different stimulation settings. However, during optimization, an algorithm can apply the same or similar stimulation settings sequentially. It is possible that interaction between sequential stimulation settings may have been obscured when randomizing between stimulation, but a confound when optimizing *in vivo*. If detected, this could have been addressed by increasing the time between stimulation when practical, or by using an alternative optimization approach that takes into account the history of stimulation settings evaluated.

7.1.4.2 Algorithm refinement

The other benefit of validating the optimization algorithm *in vivo* is the collection of additional data that can be used to further refine the algorithm. In terms of parameter space coverage, using data collected with uniform random search or an active learning approach would be most effective. However, data collected during previous optimization

experiments was useful in Chapter 5. Moreover, data collected during optimization could identify any interactions between subsequent stimulation settings as described in the previous section.

7.2 The framework in perspective

7.2.1 Engineering perspectives

The overall approach of this framework is fairly standard in more established fields of engineering such as electrical and aerospace. For example, autopilot controls can be designed by evaluating different candidate control algorithms in simulation before validating them in realistic conditions [99]. However, the simulations used in these fields of engineering are based on well-understood physical laws that can be accurately modeled to an arbitrary resolution. In contrast, the biophysical models used in neuroscience are of varying resolutions and limited accuracy. This is particularly the case for models of brain stimulation, and may result in challenges when translating the optimization approaches designed *in silico*. Rather than building simulation models from biophysical first principles, the framework described here circumvents these issues by using statistical, data-driven simulation models. While potentially requiring upfront costs to collect the initial data set, the resultant optimization algorithms perform as predicted, and efficient solutions were designed for a range of different neural modulation problems. Of course, a limitation of using a statistical model to interpolate between collected data points is that the resultant models do not reliably extrapolate and therefore cannot identify novel parameter spaces.

7.2.1.1 A tool for interpreting optimization performance

A key advantage of using a simulation based design framework such as the one described here, is the ability to anticipate results and diagnose failures. This was valuable when identifying the optimal stimulation setting that modulated hippocampal gamma power to a setpoint in Chapter 3. Prior to modeling the collected data, applying the setpoint transformation and performing the simulations, it was not apparent that the solution to the optimization problem would be a set of equivalent stimulation settings forming a ridge across the parameter space. Knowing this helped explain the behavior of the search trajectory as the estimated optimal stimulation setting would alternate between distant regions of the parameter space. Similarly, the simulations in this framework helped highlight the importance of selecting the correct algorithm configurations. When maximizing hippocampal gamma power in Chapter 2, the simulation framework accurately predicted and explained the difference in performance between the UCB(0.4) and UCB(0.01) versions of the Bayesian optimization algorithm. Ultimately, this design framework can serve as a valuable tool for predicting and understanding the behavior of a data-driven optimization algorithm when deployed in practice.

7.2.1.2 Relationship to control and reinforcement learning

Optimization is just one part of the broader engineering toolbox that can be used for delivering effective brain stimulation interventions. In particular, optimal control and reinforcement learning are potential approaches for learning a policy to determine stimulation when the effect of stimulation is dependent on the underlying state of the neural

system. While the majority of the applications in this dissertation assumed that the objective function was static, in Chapter 4 when medial septum optogenetic stimulation was applied at varying depths of anesthesia, it was shown that the effect of stimulation was influenced by anesthetic level. To address this, the pre-stimulation state was measured and added as a predictor for the surrogate Gaussian process model used in Bayesian optimization, and the predictions were conditioned on this state. While this approach was effective and a natural extension of the previously developed optimization tools, controls and RL based approaches could readily be used, and have seen some applications in neural modulation [100], [101]. However, these approaches, while powerful, still need to be designed and can be even more complex than a static optimization approach, similarly benefitting from a design framework based on statistical models derived from subject data [102].

7.2.2 Basic science perspectives

7.2.2.1 Tighter control of neural modulation experiments

Neural modulation has become a valuable tool to study functional properties of neural circuits and how brain activity relates to behavior. However, it can be important to account for subject variability and state-dependent effects of stimulation when conducting experiments using neural modulation. For example, in Chapter 3, Section 3.3.1, the medial septum optogenetic stimulation settings needed to modulate hippocampal gamma power to a given set point varied from between subjects. Similarly, in Chapter 4, the effect of medial septum optogenetic stimulation on hippocampal gamma power can be influenced by the

underlying state of the brain – in that case, depth of anesthesia. This has also been seen in relation to locomotive activity of rats undergoing medial septum optogenetic stimulation [103]. Correct application of max/minimization, setpoint, and state-dependent optimization has the potential to more accurately modulate neural state. Integrating these approaches is ultimately a stepping stone towards more sophisticated optimal control approaches that may be applicable to more challenging neural modulation problem.

7.2.2.2 Automated design of optimization solutions

Another key consideration is the accessibility of optimization as a tool. Like with any tool, experimenters would want an effective and reliable optimization solution as "off-the-shelf" as possible, avoiding the need to manually design and compare different configurations. There are several approaches to designing a bespoke optimization solution. The simplest is automatic benchmarking against a fixed battery of algorithms. This was the approach used in the applications of this framework. Specifically, a fixed grid of algorithm configurations was evaluated, except for in Chapter 5, where a uniform random search was used to determine which algorithm configurations to prototype in simulation.

This can be further built-upon by directly optimizing the configuration of the optimization algorithm. From the perspective of the design framework, the input space would be the algorithm configuration and the output objective would be the performance of the configuration on a set of simulation models. This approach may become necessary as optimization algorithms and neural modulation problems become more complex, and benchmarking many different algorithms becomes impractical. Going one step further, the

concept of ‘optimizing the optimizer’ has extended to training recurrent neural networks as general purpose solvers for sequential sampling optimization problems [98]. This more general approach may lead the way for optimization solutions for neural modulation that can perform well off-the-self in a variety of applications.

7.2.3 Clinical perspectives

7.2.3.1 Integration with clinical practice

While the technology for data-driven optimization is well-established in other fields and shows promise for neural modulation, one of the last hurdles will be integrating the algorithms with clinical practice. Given the already busy clinical schedule for treating DBS patients, it will be critical to integrate optimization solutions within the existing workflow. In the multi-objective approach detailed in Chapter 6, the optimization would take place during the procedure to implant the clinical device. However, this approach does temporarily interrupt the surgery and so additional considerations are necessary to minimize the optimization time or to develop alternative biomarkers that can be measured outside of the operating room.

A recent approach used an accelerometer in real-time to detect tremor-related symptoms while automatically adjusting stimulation under the supervision of a clinician [27]. This has the advantage of replacing an existing component of the clinical workflow with an automated process. However, some symptoms and side effects such as freezing of gait or dysarthria may be more difficult to automatically measure and detect. An alternative approach is to design the optimization algorithm as a recommender system that uses the

observations of the clinician to suggest stimulation settings to guide the programming process interactively. A proof-of-concept software application is available at: https://github.com/Precision-Neurodynamics/optimization_assistant.

7.3 Final remarks

The focus of this work was designing data-driven optimization algorithms for real-time use based on statistical models fit to subject data. However, there is a growing body of work using complex biophysical models to optimize stimulation parameters and policies, and even to identify novel types of stimulation paradigms [29], [65], [94]. While both of these approaches work towards the same goal of better delivering neural modulation, they are not mutually exclusive and can serve to complement each other. The offline approaches can be useful for *a priori* identifying regions of the parameter space with a high likelihood of producing the desired effect, while the online approaches can further refine the parameter space based on direct interaction with the subject. Continued development of both approaches will be critical to improving neural modulation as a clinical and scientific tool.

REFERENCES

[1] A.-L. Benabid *et al.*, “Subthalamic stimulation for Parkinson ’ s disease .,” vol. 31, p. 11036179, 2000.

[2] A. L. Benabid *et al.*, “Long-term suppression of tremor by chronic stimulation of the ventral intermediate thalamic nucleus.,” *Lancet (London, England)*, vol. 337, no. 8738, pp. 403–406, Feb. 1991.

[3] V. Salanova *et al.*, “Long-term efficacy and safety of thalamic stimulation for drug-resistant partial epilepsy.,” *Neurology*, vol. 84, no. 10, pp. 1017–25, Mar. 2015.

[4] T. L. Skarpaas, B. Jarosiewicz, and M. J. Morrell, “Brain-responsive neurostimulation for epilepsy (RNS® System),” *Epilepsy Res.*, vol. 153, pp. 68–70, 2019.

[5] P. E. Holtzheimer *et al.*, “Subcallosal cingulate deep brain stimulation for treatment-resistant depression: a multisite, randomised, sham-controlled trial.,” *The lancet. Psychiatry*, vol. 4, no. 11, pp. 839–849, Nov. 2017.

[6] P. Ooms, M. Mantione, M. Figee, P. R. Schuurman, P. van den Munckhof, and D. Denys, “Deep brain stimulation for obsessive-compulsive disorders: long-term analysis of quality of life.,” *J. Neurol. Neurosurg. Psychiatry*, vol. 85, no. 2, pp. 153–8, 2014.

[7] W. Xu *et al.*, “Deep brain stimulation for Tourette’s syndrome,” *Transl. Neurodegener.*, vol. 9, no. 1, p. 4, 2020.

[8] L. Grosenick, J. H. Marshel, and K. Deisseroth, “Closed-Loop and Activity-Guided Optogenetic Control,” *Neuron*, vol. 86, no. 1, pp. 106–139, 2015.

[9] A. Banaszkiewicz *et al.*, “The role of the superior parietal lobule in lexical processing of sign language: Insights from fMRI and TMS,” *Cortex*, vol. 135, pp. 240–254, 2021.

[10] A. Khan, X. Wang, C. H. E. Ti, C. Y. Tse, and K. Y. Tong, “Anodal Transcranial

Direct Current Stimulation of Anterior Cingulate Cortex Modulates Subcortical Brain Regions Resulting in Cognitive Enhancement," *Front. Hum. Neurosci.*, vol. 14, no. December, pp. 1–12, 2020.

[11] M. J. Armstrong and M. S. Okun, "Diagnosis and Treatment of Parkinson Disease: A Review.," *JAMA*, vol. 323, no. 6, pp. 548–560, Feb. 2020.

[12] J. Volkmann, J. Herzog, F. Kopper, and G. Deuschl, "Introduction to the programming of deep brain stimulators," *Mov. Disord.*, vol. 17, no. S3, pp. S181–S187, Mar. 2002.

[13] M. Picillo, A. M. Lozano, N. Kou, R. Puppi Munhoz, and A. Fasano, "Programming Deep Brain Stimulation for Parkinson's Disease: The Toronto Western Hospital Algorithms," *Brain Stimul.*, vol. 9, no. 3, pp. 425–437, 2016.

[14] A. M. Lozano *et al.*, "Deep brain stimulation: current challenges and future directions," *Nat. Rev. Neurol.*, vol. 15, no. 3, pp. 148–160, 2019.

[15] L. Perestelo-Pérez, A. Rivero-Santana, J. Pérez-Ramos, P. Serrano-Pérez, J. Panetta, and P. Hilarion, "Deep brain stimulation in Parkinson's disease: meta-analysis of randomized controlled trials," *J. Neurol.*, vol. 261, no. 11, pp. 2051–2060, 2014.

[16] C. Pollo *et al.*, "Directional deep brain stimulation: an intraoperative double-blind pilot study.," *Brain*, vol. 137, no. Pt 7, pp. 2015–2026, Jul. 2014.

[17] N. C. Swann *et al.*, "Adaptive deep brain stimulation for Parkinson's disease using motor cortex sensing," *J. Neural Eng.*, vol. 15, no. 4, p. 46006, Aug. 2018.

[18] A. T. Connolly *et al.*, "A Novel Lead Design for Modulation and Sensing of Deep Brain Structures.," *IEEE Trans. Biomed. Eng.*, vol. 63, no. 1, pp. 148–57, Jan. 2016.

[19] K. Hunka, O. Suchowersky, S. Wood, L. Derwent, and Z. H. T. Kiss, "Nursing time to program and assess deep brain stimulators in movement disorder patients.," *J. Neurosci. Nurs.*, vol. 37, no. 4, pp. 204–210, 2005.

[20] E. Keogh and A. Mueen, "Curse of Dimensionality BT - Encyclopedia of Machine Learning and Data Mining," C. Sammut and G. I. Webb, Eds. Boston, MA: Springer

US, 2017, pp. 314–315.

[21] R. S. Fisher *et al.*, “Electrical stimulation of the anterior nucleus of thalamus for treatment of refractory epilepsy,” *Epilepsia*, vol. 51, no. 5, pp. 899–908, May 2010.

[22] E. B. Geller *et al.*, “Brain-responsive neurostimulation in patients with medically intractable mesial temporal lobe epilepsy.,” *Epilepsia*, vol. 58, no. 6, pp. 994–1004, Jun. 2017.

[23] A. Cukiert, C. M. Cukiert, J. A. Burattini, P. P. Mariani, and D. F. Bezerra, “Seizure outcome after hippocampal deep brain stimulation in patients with refractory temporal lobe epilepsy: A prospective, controlled, randomized, double-blind study,” *Epilepsia*, vol. 58, no. 10, pp. 1728–1733, 2017.

[24] Z. H. T. Kiss, K. Doig-Beyaert, M. Eliasziw, J. Tsui, A. Haffenden, and O. Suchowersky, “The Canadian multicentre study of deep brain stimulation for cervical dystonia,” *Brain*, vol. 130, no. 11, pp. 2879–2886, 2007.

[25] S. K. Conroy and P. E. Holtzheimer, “Personalized Medicine in Psychiatry Deep brain stimulation for treatment-resistant depression : Predicting response and optimizing treatment,” *Pers. Med. Psychiatry*, vol. 17–18, no. November, pp. 43–45, 2019.

[26] S. F. Lempka *et al.*, “Randomized clinical trial of deep brain stimulation for poststroke pain.,” *Ann. Neurol.*, vol. 81, no. 5, pp. 653–663, May 2017.

[27] A. Haddock, K. T. Mitchell, A. Miller, J. L. Ostrem, H. J. Chizeck, and S. Miocinovic, “Automated Deep Brain Stimulation Programming for Tremor.,” *IEEE Trans. neural Syst. Rehabil. Eng. a Publ. IEEE Eng. Med. Biol. Soc.*, vol. 26, no. 8, pp. 1618–1625, Aug. 2018.

[28] F. Steigerwald, L. Müller, S. Johannes, C. Matthies, and J. Volkmann, “Directional deep brain stimulation of the subthalamic nucleus: A pilot study using a novel neurostimulation device,” *Mov. Disord.*, vol. 31, no. 8, pp. 1240–1243, 2016.

[29] D. T. Brocker, B. D. Swan, R. Q. So, D. A. Turner, R. E. Gross, and W. M. Grill, “Optimized temporal pattern of brain stimulation designed by computational evolution,” *Sci. Transl. Med.*, vol. 9, no. 371, p. eaah3532, 2017.

[30] K. Kumaravelu, C. S. Oza, C. E. Behrend, and W. M. Grill, “Model-based deconstruction of cortical evoked potentials generated by subthalamic nucleus deep brain stimulation,” *J Neurophysiol*, vol. 120, pp. 662–680, 2018.

[31] F. Mina *et al.*, “Model-guided control of hippocampal discharges by local direct current stimulation,” *Sci. Rep.*, vol. 7, no. 1, p. 1708, 2017.

[32] Y. Denoyer, I. Merlet, F. Wendling, and P. Benquet, “Modelling acute and lasting effects of tDCS on epileptic activity,” *J. Comput. Neurosci.*, vol. 48, no. 2, pp. 161–176, 2020.

[33] J. Stiso *et al.*, “White Matter Network Architecture Guides Direct Electrical Stimulation through Optimal State Transitions,” *Cell Rep.*, vol. 28, no. 10, pp. 2554-2566.e7, 2019.

[34] B. Howell, K. Gunalan, and C. C. McIntyre, “A Driving-Force Predictor for Estimating Pathway Activation in Patient-Specific Models of Deep Brain Stimulation,” *Neuromodulation*, vol. 22, no. 4, pp. 403–415, 2019.

[35] R. Lorenz *et al.*, “The Automatic Neuroscientist : automated experimental design with real-time fMRI,” *Neuroimage*, vol. 129, pp. 1–22, 2016.

[36] J. R. Gardner, X. Song, K. Q. Weinberger, D. Barbour, and J. P. Cunningham, “Psychophysical detection testing with Bayesian active learning,” *Uncertain. Artif. Intell. - Proc. 31st Conf. UAI 2015*, pp. 286–295, 2015.

[37] M. J. Connolly *et al.*, “A framework for designing data-driven optimization systems for neural modulation,” *J. Neural Eng.*, 2020.

[38] S.-E. Park *et al.*, “A Machine Learning Approach to Characterize the Modulation of the Hippocampal Rhythms Via Optogenetic Stimulation of the Medial Septum,” *Int. J. Neural Syst.*, vol. 29, no. 10, p. 1950020, Dec. 2019.

[39] N. G. Laxpati, B. Mahmoudi, C.-A. Gutekunst, J. P. Newman, R. Zeller-Townson, and R. E. Gross, “Real-time in vivo optogenetic neuromodulation and multielectrode electrophysiologic recording with NeuroRighter,” *Front. Neuroeng.*, vol. 7, no. October, pp. 1–15, 2014.

[40] N. G. Laxpati, B. Mahmoudi, C.-A. Gutekunst, J. P. Newman, R. Zeller-Townson, and R. E. Gross, “Real-time in vivo optogenetic neuromodulation and multielectrode electrophysiologic recording with NeuroRighter,” *Front. Neuroeng.*, vol. 7, no. October, pp. 1–15, 2014.

[41] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*. MIT Press, 2006.

[42] J. C. Spall, “Implementation of the simultaneous perturbation algorithm for stochastic optimization,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 34, no. 3, pp. 817–823, Jul. 1998.

[43] P.-T. de Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, “A Tutorial on the Cross-Entropy Method,” *Ann. Oper. Res.*, vol. 134, no. 1, pp. 19–67, Feb. 2005.

[44] E. Brochu, V. M. Cora, and N. De Freitas, “A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning,” *Arxiv Prepr. arXiv1012.2599*, pp. 1–49, 2010.

[45] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*. MIT Press, 2006.

[46] F. Hutter, H. Hoos, and K. Leyton-Brown, “Sequential Model-Based Optimization for General Algorithm Configuration Lecture Notes in Computer Science,” *Mach. Learn. Knowl. Discov. Databases*, no. Chapter 40, pp. 507–523, 2011.

[47] A. Kendall and Y. Gal, “What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?,” no. Nips, 2017.

[48] M. J. Connolly, S.-E. Park, and R. E. Gross, “Learning State-Dependent Neural Modulation Policies with Bayesian Optimization,” *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, pp. 6454–6457, 2019.

[49] M. K. Van Vugt, A. Schulze-Bonhage, B. Litt, A. Brandt, and M. J. Kahana, “Hippocampal gamma oscillations increase with memory load,” *J. Neurosci.*, vol. 30, no. 7, pp. 2694–2699, 2010.

[50] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998.

[51] N. Grossman, K. Nikolic, M. S. Grubb, J. Burrone, C. Toumazou, and P. Degenaar, “High-frequency limit of neural stimulation with ChR2.,” *Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. IEEE Eng. Med. Biol. Soc. Annu. Int. Conf.*, vol. 2011, pp. 4167–4170, 2011.

[52] P. Feliot, J. Bect, and E. Vazquez, “A Bayesian approach to constrained single-and multi-objective optimization.”

[53] F. Berkenkamp, R. Moriconi, A. P. Schoellig, and A. Krause, “Safe Learning of Regions of Attraction for Uncertain, Nonlinear Systems with Gaussian Processes,” 2016.

[54] J. Gonzalez, Z. Dai, A. Damianou, and N. D. Lawrence, “Preferential Bayesian Optimization,” *Arxiv*, 2017.

[55] K. J. van Dijk, R. Verhagen, L. J. Bour, C. Heida, and P. H. Veltink, “Avoiding Internal Capsule Stimulation With a New Eight-Channel Steering Deep Brain Stimulation Lead,” *Neuromodulation Technol. Neural Interface*, vol. 21, no. 6, pp. 553–561, Aug. 2018.

[56] A. J. Romann, B. C. Beber, C. A. Cielo, and C. R. de Mello Rieder, “Acoustic voice modifications in individuals with Parkinson disease submitted to deep brain stimulation,” *Int. Arch. Otorhinolaryngol.*, vol. 23, no. 2, pp. 203–208, 2019.

[57] S. Reuter, G. Deuschl, D. Berg, A. Helmers, D. Falk, and K. Witt, “Life-threatening DBS withdrawal syndrome in Parkinson’s disease can be treated with early reimplantation,” *Parkinsonism Relat. Disord.*, vol. 56, pp. 88–92, Nov. 2018.

[58] M. Turchetta, F. Berkenkamp, and A. Krause, “Safe Exploration for Interactive Machine Learning,” *33rd Conf. Neural Inf. Process. Syst.*, 2019.

[59] M. Turchetta, F. Berkenkamp, and A. Krause, “Safe exploration for interactive machine learning,” *arXiv*, no. 2014, 2019.

[60] S. A. Desai, J. D. Rolston, C. E. McCracken, S. M. Potter, and R. E. Gross, “Asynchronous Distributed Multielectrode Microstimulation Reduces Seizures in the Dorsal Tetanus Toxin Model of Temporal Lobe Epilepsy.,” *Brain Stimul.*, vol. 9, no. 1, pp. 86–100, Jan. 2015.

[61] O. Ashmaig, M. J. Connolly, R. E. Gross, and B. Mahmoudi, “Bayesian Optimization of Asynchronous Distributed Microelectrode Theta Stimulation and Spatial Memory,” *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, vol. 2018-July, pp. 2683–2686, 2018.

[62] M. Antunes and G. Biala, “The novel object recognition memory: Neurobiology, test procedure, and its modifications,” *Cogn. Process.*, vol. 13, no. 2, pp. 93–110, 2012.

[63] M. J. Connolly *et al.*, “Multi-Objective Data-Driven Optimization for Improving Deep Brain Stimulation in Parkinson’s Disease,” *J. Neural Eng.*, 2021.

[64] R. Lorenz *et al.*, “Efficiently searching through large tACS parameter spaces using closed-loop Bayesian optimization,” *Brain Stimul.*, no. xxxx, 2019.

[65] L. L. Grado, M. D. Johnson, and T. I. Netoff, “Bayesian adaptive dual control of deep brain stimulation in a computational model of Parkinson’s disease,” *PLoS Comput. Biol.*, vol. 14, no. 12, pp. 1–23, 2018.

[66] Y. Sui, vincent Zhuang, J. Burdick, and Y. Yue, “Stagewise Safe {B}ayesian Optimization with {G}aussian Processes,” in *Proceedings of the 35th International Conference on Machine Learning*, 2018, vol. 80, pp. 4781–4789.

[67] B. Letham, R. Calandra, A. Rai, and E. Bakshy, “Re-examining linear embeddings for high-dimensional Bayesian Optimization,” *arXiv*, no. NeurIPS, 2020.

[68] A. D. Kirsch, S. Hassin-Baer, C. Matthies, J. Volkmann, and F. Steigerwald, “Anodic versus cathodic neurostimulation of the subthalamic nucleus: A randomized-controlled study of acute clinical effects.,” *Parkinsonism Relat. Disord.*, vol. 55, pp. 61–67, Oct. 2018.

[69] J. L. Vitek *et al.*, “Subthalamic nucleus deep brain stimulation with a multiple independent constant current-controlled device in Parkinson’s disease

(INTREPID): a multicentre, double-blind, randomised, sham-controlled study.," *Lancet. Neurol.*, vol. 19, no. 6, pp. 491–501, Jun. 2020.

[70] P. Krack, V. Fraix, A. Mendes, A.-L. Benabid, and P. Pollak, "Postoperative management of subthalamic nucleus stimulation for Parkinson's disease," *Mov. Disord.*, vol. 17, no. S3, pp. S188–S197, Mar. 2002.

[71] I. Iturrate *et al.*, "Beta-driven closed-loop deep brain stimulation can compromise human motor behavior in Parkinson's Disease," *bioRxiv*, p. 696385, Jan. 2019.

[72] L. A. Johnson, S. D. Nebeck, A. Muralidharan, M. D. Johnson, K. B. Baker, and J. L. Vitek, "Closed-Loop Deep Brain Stimulation Effects on Parkinsonian Motor Symptoms in a Non-Human Primate - Is Beta Enough?," *Brain Stimul.*, vol. 9, no. 6, pp. 892–896, 2016.

[73] L. A. Johnson, J. Wang, S. D. Nebeck, J. Zhang, M. D. Johnson, and J. L. Vitek, "Direct activation of primary motor cortex during subthalamic but not pallidal deep brain stimulation," *J. Neurosci.*, vol. 40, no. 10, pp. 2166–2177, 2020.

[74] B. Rosin *et al.*, "Closed-Loop Deep Brain Stimulation Is Superior in Ameliorating Parkinsonism," *Neuron*, vol. 72, no. 2, pp. 370–384, 2011.

[75] S. Little *et al.*, "Adaptive deep brain stimulation for Parkinson's disease demonstrates reduced speech side effects compared to conventional stimulation in the acute setting," *J. Neurol. Neurosurg. Psychiatry*, vol. 87, no. 12, pp. 1388–1389, Dec. 2016.

[76] S. Miocinovic *et al.*, "Cortical potentials evoked by subthalamic stimulation demonstrate a short latency hyperdirect pathway in humans," *J. Neurosci.*, vol. 38, no. 43, pp. 1327–18, 2018.

[77] P. Ashby, Y. J. Kim, R. Kumar, A. E. Lang, and A. M. Lozano, "Neurophysiological effects of stimulation through electrodes in the human subthalamic nucleus," *Brain*, vol. 122, no. 10, pp. 1919–1931, 1999.

[78] P. Mahlknecht *et al.*, "Pyramidal tract activation due to subthalamic deep brain stimulation in Parkinson's disease," *Mov. Disord.*, vol. 32, no. 8, pp. 1174–1182, 2017.

[79] J. Costa, J. Valls-Solé, F. Valldeoriola, J. Rumià, and E. Tolosa, “Motor responses of muscles supplied by cranial nerves to subthalamic nucleus deep brain stimuli,” *Brain*, vol. 130, no. 1, pp. 245–255, 2007.

[80] P. Feliot, J. Bect, and E. Vazquez, “A Bayesian approach to constrained single- and multi-objective optimization,” *J. Glob. Optim.*, vol. 67, no. 1, pp. 97–133, 2017.

[81] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley, 2001.

[82] S. Little *et al.*, “Adaptive deep brain stimulation in advanced Parkinson disease.,” *Ann. Neurol.*, vol. 74, no. 3, pp. 449–457, 2013.

[83] M. Malekmohammadi *et al.*, “Kinematic Adaptive Deep Brain Stimulation for Resting Tremor in Parkinson’s Disease,” *Mov. Disord.*, vol. 31, no. 3, pp. 426–428, 2016.

[84] S. Miocinovic *et al.*, “Cortical potentials evoked by subthalamic stimulation demonstrate a short latency hyperdirect pathway in humans,” *J. Neurosci.*, vol. 38, no. 43, pp. 1327–18, 2018.

[85] N. R. Costa and J. A. Lourenço, “Exploring Pareto Frontiers in the Response Surface Methodology BT - Transactions on Engineering Technologies,” 2015, pp. 399–412.

[86] N. Srinivas and K. Deb, “Muiltiobjective Optimization Using Nondominated Sorting in Genetic Algorithms,” *Evol. Comput.*, vol. 2, no. 3, pp. 221–248, 1994.

[87] M. T. M. Emmerich, A. H. Deutz, and J. W. Klinkenberg, “Hypervolume-based expected improvement: Monotonicity properties and exact computation,” in *2011 IEEE Congress of Evolutionary Computation (CEC)*, 2011, pp. 2147–2154.

[88] P. Sarikhani, S. Miocinovic, and B. Mahmoudi, “Towards automated patient-specific optimization of deep brain stimulation for movement disorders,” in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2019, pp. 6159–6162.

[89] A. Haddock, K. T. Mitchell, A. Miller, J. L. Ostrem, H. J. Chizeck, and S. Miocinovic, “Automated Deep Brain Stimulation Programming for Tremor,” *IEEE*

Trans. Neural Syst. Rehabil. Eng., vol. 26, no. 8, pp. 1618–1625, 2018.

[90] Y.-C. Wang *et al.*, "Probing circuit of Papez with stimulation of anterior nucleus of the thalamus and hippocampal evoked potentials," *Epilepsy Res.*, vol. 159, p. 106248, 2020.

[91] V. Picheny, "Multiobjective optimization using Gaussian process emulators via stepwise uncertainty reduction," *Stat. Comput.*, vol. 25, no. 6, pp. 1265–1280, 2015.

[92] E. C. Garrido-Merchán and D. Hernández-Lobato, "Predictive Entropy Search for Multi-objective Bayesian Optimization with Constraints," *Neurocomputing*, vol. 361, pp. 50–68, 2019.

[93] W. Ponweiser, T. Wagner, D. Biermann, and M. Vincze, "Multiobjective Optimization on a Limited Budget of Evaluations Using Model-Assisted S-Metric Selection BT," in *International Conference on Parallel Problem Solving from Nature*, 2008, pp. 784–794.

[94] E. Peña *et al.*, "Multi-objective particle swarm optimization for postoperative deep brain stimulation targeting of subthalamic nucleus pathways.," *J. Neural Eng.*, vol. 15, no. 6, p. 66020, Dec. 2018.

[95] Y. Sui, Y. Yue, and J. W. Burdick, "Correlational dueling bandits with application to clinical treatment in large decision spaces," *arXiv*, 2017.

[96] M. Jakobs *et al.*, "A multicenter, open-label, controlled trial on acceptance, convenience, and complications of rechargeable internal pulse generators for deep brain stimulation: the Multi Recharge Trial.," *J. Neurosurg.*, pp. 1–9, Aug. 2019.

[97] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, 2012.

[98] Y. Chen *et al.*, "Learning to learn without gradient descent by gradient descent," *34th Int. Conf. Mach. Learn. ICML 2017*, vol. 2, pp. 1252–1260, 2017.

[99] J. Pravitra, E. Theodorou, and E. N. Johnson, "Flying Complex Maneuvers with Model Predictive Path Integral Control," in *AIAA Scitech 2021 Forum*, American

Institute of Aeronautics and Astronautics, 2021.

[100] M. F. Bolus, A. A. Willats, C. J. Whitmire, C. J. Rozell, and G. B. Stanley, “Design strategies for dynamic closed-loop optogenetic neurocontrol in vivo,” *J. Neural Eng.*, vol. 15, no. 2, 2018.

[101] S. S. Kumar, J. Boedecker, M. Riedmiller, and U. Egert, “Adaptive Long-term Control of Biological Neural Networks with Deep Reinforcement Learning,” *Neurocomputing*, no. 2019, 2018.

[102] Y. Yang *et al.*, “Modelling and prediction of the dynamic responses of large-scale brain networks during direct electrical stimulation,” *Nat. Biomed. Eng.*, 2021.

[103] B. J. Blumberg *et al.*, “Efficacy of nonselective optogenetic control of the medial septum over hippocampal oscillations: The influence of speed and implications for cognitive enhancement,” *Physiol. Rep.*, vol. 4, no. 23, pp. 1–20, 2016.